RADIOACTIVE WATERMARKING OF GRAPH NODE CLASSIFICATION DATASETS FOR SCALABLE OWNER ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph datasets used in industrial graph neural network (GNN) applications are valuable proprietary assets, often licensed to multiple buyers under restricted-use agreements. When a suspect GNN trained on a buyer's leaked dataset copy appears, the owner must identify the responsible buyer, since only buyer-level attribution allows terminating the license and pursuing legal remedies. However, this task is unsupported by existing dataset watermarks which provide only binary ownership verification and fail under multi-buyer owner tracing.

In this paper, we introduce the first *radioactive* watermarking method for graph node classification datasets that supports *scalable* owner attribution. Radioactive means that the watermark signal persists through training and is recoverable from model outputs, without access to the released data, the buyer's training procedure, or the GNN architecture. Detection is therefore *architecture-agnostic* and black-box, meaning it can be applied to suspect models with different architectures, and the underlying test is a closed-form statistic with a provably bounded false-positive rate, jointly protecting dataset and model intellectual property. We attribute a leaked suspect to its true buyer at marketplace scale via a per-buyer watermarking test, with a closed-form attribution rate that saturates with the number of buyers.

Across five benchmarks and eight GNN architectures, our method achieves a stealthy yet highly detectable watermark signal that remains robust under cross-architecture detection and four removal attacks, while incurring negligible degradation in task accuracy, and provides reliable owner traceability at the scale of multi-buyer data marketplaces. Our code and data are available at `https://anonymous.4open.science/r/graph_watermark-0C2F/README.md`.

1 INTRODUCTION

Graph neural networks (GNNs) are increasingly deployed in high-value industrial settings including recommendation, fraud detection, and social analysis, where the underlying graph datasets are proprietary assets built from domain expertise, manual annotation, and real-world relationship mining. These datasets are now monetized on data marketplaces (e.g., AWS Data Exchange (Amazon Web Services, 2026), Snowflake Marketplace (Snowflake Inc., 2026)) under *restricted-use, non-sublicensable* licenses: each buyer may train models on its copy but may not redistribute the copy. When a suspect GNN trained on a buyer's leaked copy appears without authorization, the dataset owner has access only to this model, since the copy is licensed not to be redistributed and remains outside the owner's reach. Identifying *which buyer's copy* was used is essential because only buyer-level attribution allows the owner to terminate the specific licensing contract and pursue legal remedies against the leaker, without which the license becomes unenforceable in practice.

We refer to a dataset's watermark as *radioactive* if its signal survives downstream training and remains statistically detectable in the trained model, without re-accessing the released data. Radioactivity has been demonstrated in single-buyer dataset watermarks for image (Sablayrolles et al., 2020; Bouaziz et al., 2025), text (Rastogi et al., 2025), LLM output (Sander et al., 2024), and tabular (Wu et al., 2024) data. Existing multi-buyer traceability schemes (Boneh & Shaw, 1998; Guo et al., 2006; Zheng et al., 2025; Cohen et al., 2025) verify from the released copy or inject watermarks at inference time,

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.

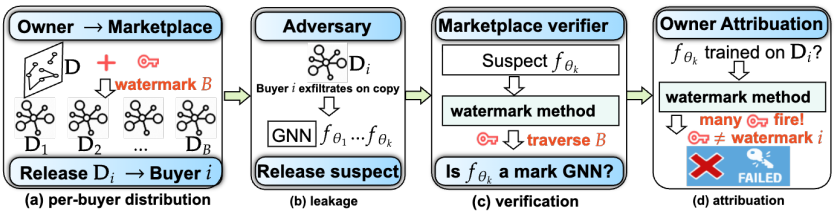


Figure 1: **Owner attribution failure of trigger-based watermarks in a graph-data marketplace.**

not by recovering the watermark from a downstream-trained model alone. *Owner attribution* for graph data remains unaddressed: existing graph-data schemes are trigger-based and provide only binary ownership verification, supporting neither radioactive model-only detection nor scalable owner attribution across multiple buyers. Therefore, **we introduce the first *radioactive* dataset watermark for graph node classification that supports *scalable* owner attribution, bringing the radioactivity framework to graph data and enabling per-buyer leaker identification from the model alone.**

Existing graph data watermarks (e.g., BkdOwn (Li et al., 2023), Backdoor (Xu et al., 2023)) cannot scale to owner attribution: their triggers activate overlapping feature subspaces in the trained GNN, so a model trained on one buyer's copy still responds to foreign triggers, a failure we term *cross-fire* (Figure 1; empirical evidence in Table 25). The root cause is the structural non-orthogonality of the triggers. A secret key alone (e.g., KGMark (Peng et al., 2025)) is *necessary but not sufficient*: it must additionally parameterize *mutually independent, non-interfering watermark carriers* that occupy distinct subspaces and never appear in the released data, the design principle our method enforces. Appendix B details additional design gaps for node classification.

Scalable owner attribution requires watermark carriers occupying near-orthogonal directions in embedding space. Fixed-pattern triggers, whether structural or feature-based, lack such orthogonality: their response regions overlap across buyers and cross-fire as the pool grows. *Feature perturbation*, guided by a principled optimization objective, embeds each key direction into the model's internal representations. Feature is preferred over structural: adversarial-attack studies (Zügner et al., 2018; Zügner & Günnemann, 2019) show that message passing amplifies adjacency perturbations, making them more damaging and easier to sanitize, whereas feature perturbations leave the topology intact and resist isolation without harming utility. The same studies supply the optimization machinery for crafting minimal, imperceptible changes that shape model behavior.

Watermark injection should not alter the model's predictions, because buyers train on the released data as their training set. This inverts the adversarial-attack objective: whereas attacks seek minimal perturbations that *change* predictions, our watermarking scheme seeks minimal perturbations that *preserve* predictions while embedding a persistent, detectable signal. The signal must survive buyer training so the owner can verify ownership on the suspect model. We thus optimize four properties: (i) **stealthiness** ensures imperceptible feature perturbations, (ii) low **harmfulness** preserves predictions on watermarked nodes, (iii) **detectability** ensures reliable signal absorption by the model, and (iv) **robustness** ensures signal persistence under fine-tuning, pruning, and architecture changes.

Contributions. **(i) Radioactive watermark for scalable owner attribution.** We propose the first *radioactive* graph node classification dataset watermark for *scalable* owner attribution, with *architecture-agnostic* verification from the trained model alone. **(ii) Direction-guided feature perturbation with mutually independent watermark carriers.** We design an objective-guided optimization so that the watermark survives model training and is absorbed into the model's weights, without label poisoning or training control. Buyer-specific carriers occupy distinct embedding subspaces, enabling scalable owner attribution. **(iii) Closed-form detection guarantee and attribution estimate.** At watermark-injection time, we design a signal estimator that predicts and certifies at $(1-\delta_B)$-probability whether the watermark will remain detectable in the future suspect model. We also provide a closed-form estimate of the owner attribution success rate across B buyers.

2 Preliminaries

2.1 Node Classification and Graph Neural Networks

Node Classification in a Graph. We consider node classification in a graph $G = (V, E, \mathbf{X})$ with nodes V, edges $E \subseteq \{(v, w) \mid v, w \in V\}$, and node features $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{|V|-1}\}$ ($\mathbf{x}_i \in \mathbb{R}^d$).

The adjacency matrix $\mathbf{A} \in \{0,1\}^{|V|\times|V|}$ has entries $\mathbf{A}_{vw} = 1$ if $(v, w) \in E$ and 0 otherwise. A labeled subset $V_L \subseteq V$ forms the training data; each labeled node has a class $y_v \in C = \{1, \ldots, c\}$. Given $\mathbf{A}$ and $\mathbf{X}$, the goal is to predict the label of each test node $v \in V \setminus V_L$. Notations are provided in Table 3 in the Appendix.

Graph Neural Networks. GNNs classify nodes via message passing (Kipf & Welling, 2017): each node's representation is iteratively updated by aggregating and transforming information from its neighbors. We use the Graph Convolutional Network (GCN) as the backbone for concreteness in this section; §3 uses GCN as the owner's design-time surrogate, while suspect models evaluated in §4 may use any GNN architecture. We write $\mathbf{H}^{(l)}$ for the hidden node embeddings at layer l, with $\mathbf{H}^{(0)} = \mathbf{X}$ and $\mathbf{H}^{(K)}$ at the final layer. A softmax classifier on $\mathbf{H}^{(K)}$ yields predicted labels $\hat{y}_v$.

2.2 Marketplace Setup

Data owner, buyers, licensing. In a data marketplace, the data owner $\mathcal{V}$ owns a benign dataset $\mathcal{D} = (G, Y, V_L)$ with graph $G = (V, E, \mathbf{X})$, labels $Y \in \{1, \ldots, c\}^{|V|}$, and training set V_L. The owner licenses $\mathcal{D}$ to B buyers $\{b_1, \ldots, b_B\}$ under a *restricted-use, non-sublicensable* agreement: each may train its own model but may not redistribute the copy. To enable owner attribution after a leak, the owner releases a personalised watermarked copy $\mathcal{D}^{(i)}_{\text{wm}}$ to each b_i under a buyer-specific private key $\mathcal{K}_i$. The watermark is *detective* and post-hoc: after an unauthorized model release, it identifies the *leaker* rather than preventing the release. The leaker is the buyer whose copy was used to train the released model. Prevention via legal or access-control measures is out of scope for this work.

Private key and carrier signal. The **private key** $\mathcal{K}_i = \{u^{(i)}_c\}^C_{c=1} \subset \mathbb{R}^d$ consists of C independently sampled unit vectors held exclusively by the owner. Each $u^{(i)}_c$ is a per-class carrier direction toward which watermark injection drives the embeddings. The resulting alignment imprinted in the trained weights is the **carrier signal**, detectable only by the holder of $\mathcal{K}_i$.

Threat model. A suspect model f_θ trained on some buyer b_i's copy $\mathcal{D}^{(i)}_{\text{wm}}$ appears without the owner's authorization, evidencing a breach of the licensing agreement by b_i. The owner sees only f_θ and must decide (a) whether f_θ was trained on any watermarked copy, and (b) which buyer is the leaker. Two access settings below specify what the owner can probe on f_θ for *detection*. **White-Box Setting.** The owner sees f_θ's architecture and weights but not its training procedure or hyperparameters. Detection extracts the last-layer weights $\mathbf{W}_\theta$ and tests systematic alignment with each buyer's carriers $\{u^{(i)}_c\}^C_{c=1}$. **Black-Box Setting.** Only per-node predictions are exposed, as hard labels or soft outputs. Architecture and parameters are hidden. Detection uses the GNN-as-a-service inductive interface, where a node query with local subgraph context returns f_θ's output.

2.3 Problem Formulation

Given a suspect f_θ from some leaker $b_{i^\star}$ or from a benign actor never exposed to any watermarked copy, the data owner must perform: (i) **Watermark Detection.** Decide whether f_θ was trained on any of the watermarked copies $\{\mathcal{D}^{(i)}_{\text{wm}}\}^B_{i=1}$ or on a benign dataset $\mathcal{D}$. (ii) **Scalable Owner Attribution.** If watermarked, identify the leaker's index $i^\star \in \{1, \ldots, B\}$.

3 Methodology

3.1 Method Overview

We construct our dataset watermark by shifting node embeddings toward a secret direction. For each class, the owner generates a random unit vector u_c as a private key, which remains undisclosed. The watermark is injected into a small subset of nodes V_w by perturbing their features so that the resulting embedding shift aligns with u_c (Figure 2). This geometric design yields two key advantages: (i) *radioactivity*, enabling watermark detection directly from a suspect model, and (ii) *scalable traceability*, supporting reliable owner attribution across multiple buyers.

Radioactivity. Once injected into the graph dataset, the watermark signal persists through downstream training and can be recovered directly from the suspect model's output, without requiring access to

the released data, the buyer's training procedure, or the model architecture. We term the watermark *radioactive* because the embedding shift on the watermark node subset V_w is readily absorbed by any model trained on the watermarked dataset. During training, each feature perturbation on V_w induces a corresponding embedding shift, which backpropagation transfers into the per-class linear classifier head weight $\mathbf{W}_c$. This rotation aligns $\mathbf{W}_c$ with the projection of the secret direction u_c onto the shared embedding subspace S. This mechanism makes detection *architecture-agnostic*: regardless of the suspect model's architecture, $\mathbf{W}_c$ consistently aligns with the projection of u_c onto S, producing strong per-class alignment. Detection then reduces to evaluating these per-class alignments and Fisher-combining the evidence across classes, yielding a model-only test statistic.

Scalable traceability. Our watermark framework enables large-scale owner attribution because different owners' private directions u_c remain mutually distinguishable on a given suspect model. Denote B as the total number of buyers in the marketplace. This distinguishability arises because each u_c is sampled independently as a random unit vector in the high-dimensional embedding space, making different owners' u_c nearly orthogonal with high probability. As a result, foreign owners' u_c exhibit negligible alignment with $\mathbf{W}_c$, while the true owner's u_c remains strongly aligned with $\mathbf{W}_c$ inside S. Our owner attribution is therefore *scalable* with buyer count B: the abundance of near-orthogonal directions in high-dimensional space ensures that this distinguishability persists even as the marketplace grows. Traceability is realized by evaluating the alignment of $\mathbf{W}_c$ with each owner's u_c, and the strongest Fisher-combined evidence across classes identifies the true owner. We formalize these two phases in the remainder of this section, with the end-to-end pipeline shown in Figure 6 (Appendix E.1).

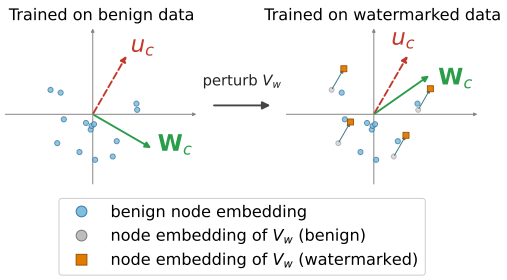


Figure 2: **Geometric intuition of our method.** In a high-dimensional space, the linear classifier head weight $\mathbf{W}_c$, which separates its class, is nearly orthogonal to u_c with high probability. Watermark injection shifts each V_w's embedding toward u_c, thereby aligning $\mathbf{W}_c$ with u_c.

3.2 Watermark Injection via Direction-Guided Feature Perturbation

Watermark injection produces the watermarked dataset $\mathcal{D}_{\text{wm}}$ in two steps: (i) selecting a watermark node subset V_w, and (ii) optimizing per-node feature perturbations on V_w so that each selected node's embedding shift $\Delta\mathbf{H}_v$ aligns with its class's private direction u_c.

Watermark Nodes Selection. We restrict the watermark to a small node subset V_w because perturbing too many nodes would distort the released dataset $\mathcal{D}$ and degrade downstream task accuracy. We select V_w by *class-balanced random sampling*, uniformly drawing a fixed number of nodes per class without replacement. This ensures that each class contributes sufficient embedding shifts for the trained model to absorb the watermark signal and produce strong per-class alignment. Alternative heuristics (e.g., graph distance, node degree, classifier confidence, or embedding diversity) are possible, since message passing propagates perturbations to neighbors of V_w, allowing the watermark signal to be absorbed from the local neighborhood rather than V_w alone (Appendix E.2).

Feature Perturbation by Optimization. Random perturbations yield arbitrary embedding shifts $\Delta\mathbf{H}_v$ unaligned with u_c. We therefore cast feature perturbation as an optimization problem: find $\Delta\mathbf{x}_v$ for each $v \in V_w$ so that $\Delta\mathbf{H}_v$ aligns with its class-specific secret direction u_{y_v}. Alignment alone is insufficient: unconstrained $\Delta\mathbf{x}_v$ may maximize alignment but (i) become trivially detectable, or (ii) induce large $\Delta\mathbf{H}_v$ that degrades classification. Thus, optimization enforces two constraints: (i) *Stealthiness*. $\Delta\mathbf{x}_v$ must remain small so watermarked features stay close to the originals, making the watermark imperceptible. (ii) *Harmfulness*. $\Delta\mathbf{H}_v$ must remain bounded so embeddings stay near their original positions, preserving predictions and limiting accuracy degradation.

Differentiable Surrogate Encoder. For each $v \in V_w$, we define $\Delta\mathbf{x}_v \in \mathbb{R}^d$ as the difference between the watermarked and original feature vectors: $\Delta\mathbf{x}_v = \mathbf{x}_{\text{wm},v} - \mathbf{x}_{\text{orig},v}$. To compute the induced embedding shift $\Delta\mathbf{H}_v$, we use a differentiable *surrogate* encoder that approximates the unknown suspect. Specifically, we adopt a single-layer linear GCN: $\mathbf{H}_s = \hat{\mathbf{A}}\mathbf{X}\mathbf{W}_s$, where $\hat{\mathbf{A}}$ is the

normalized adjacency matrix, $\mathbf{X}$ the node feature matrix, and $\mathbf{W}_s$ a learnable weight matrix matched to u_c. This design ensures: (i) *shallowness*, keeping stochastic gradient descent optimization well-conditioned via a smaller Jacobian condition number induced by lower GCN depth; (ii) *linearity*, so $\Delta\mathbf{H}_v$ depends linearly on $\Delta\mathbf{x}_v$; and (iii) *dimension matching*, with $u_c \in \mathbb{R}^{d_s}$ and $d_s \leq d$, ensuring u_c lies within the surrogate's output space. Each design choice is empirically validated in Appendix E.3. The perturbation then propagates through the surrogate as: $\Delta\mathbf{H}_v = \mathbf{H}_s(\mathbf{x}_{\mathrm{wm},v}) - \mathbf{H}_s(\mathbf{x}_{\mathrm{orig},v})$.

Loss Function. We optimize $\Delta\mathbf{x}_v$ to align $\Delta\mathbf{H}_v$ with u_{y_v} while keeping perturbations imperceptible. The loss is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{align}} + \lambda_1 \mathcal{L}_x + \lambda_2 \mathcal{L}_{\mathbf{H}} \quad \text{s.t. } \|\Delta\mathbf{x}_v\|_\infty \leq \delta,\ \forall v \in V_w, \tag{1}$$

where

- $\mathcal{L}_{\mathrm{align}} = \frac{1}{|V_w|}\sum_{v\in V_w} \arccos\big(\cos(\Delta\mathbf{H}_v, u_{y_v})\big)$ enforces **detectability** by driving embedding shifts $\Delta\mathbf{H}_v$ to align with the owner's secret direction u_{y_v}, making the watermark reliably recoverable;
- $\mathcal{L}_x = \frac{1}{|V_w|}\sum_{v\in V_w} \|\Delta\mathbf{x}_v\|_2$ enforces **stealthiness** by penalizing large feature perturbations $\Delta\mathbf{x}_v$, keeping the watermarked dataset almost indistinguishable from the original; and
- $\mathcal{L}_{\mathbf{H}} = \frac{1}{|V_w|}\sum_{v\in V_w} \big\|\Delta\mathbf{H}_v - \langle\Delta\mathbf{H}_v, u_{y_v}\rangle\, u_{y_v}\big\|_2$ enforces **harmlessness** by bounding the embedding shift orthogonal to the secret direction u_{y_v}, the part that adds no alignment signal, preserving node predictions and preventing accuracy degradation.

Optimization terminates once each per-node cosine alignment $\cos(\Delta\mathbf{H}_v, u_{y_v})$ reaches the target θ; the full procedure is given in Appendix E.1, Algorithm 1.

3.3 Radioactive Watermark Detection: Alignment Test

Watermark detection provides a formally provable alignment test between a suspect model f_θ's classifier weight $\mathbf{W}_c$ and the specified private direction u_c, verifying whether the *radioactive* signal survives in f_θ trained on the watermarked dataset $\mathcal{D}_{\mathrm{wm}}$. Two considerations shape this verification: (a) the buyer may train f_θ with *any* GNN architecture, so detection must be **architecture-agnostic**; it cannot rely on knowing or controlling the suspect's frame. (b) The watermark signal must be **quantifiable**, producing a statistically valid test with bounded false-positive rate, so that the decision is admissible as evidence rather than a heuristic.

Cross-Architecture Alignment. Detection must remain *architecture-agnostic* because suspects may train f_θ with arbitrary GNNs. The alignment test therefore relates two embedding spaces: the surrogate f_s, where the carrier u_c was defined at injection, and the suspect f_θ, where $\mathbf{W}_c$ is learned. Since these spaces differ in axis ordering and dimension, a direct comparison is inappropriate (Zhou et al., 2023; Keriven & Peyré, 2019). We bridge them via an alignment matrix $\mathbf{X}_{\mathrm{align}} \in \mathbb{R}^{d\times d_{\mathbf{H}}}$, fit by least-squares regression on N_{fit} benign nodes:

$$\mathbf{X}_{\mathrm{align}} = \arg\min_{\mathbf{X}} \|\mathbf{H}_s\mathbf{X} - \mathbf{H}_\theta\|_2^2 = \mathbf{H}_s^\dagger\mathbf{H}_\theta,$$

where $\mathbf{H}_s^\dagger$ is the pseudoinverse of the surrogate embeddings $\mathbf{H}_s$, and $\mathbf{H}_\theta$ denotes the suspect embeddings. We invert $\mathbf{H}_s$ (forward direction) rather than $\mathbf{H}_\theta$ (backward), since the shallow surrogate yields a stable singular spectrum, while deeper suspects can be ill-conditioned. Also, empirically the forward choice performs better (Appendix D.2). Vectors from the surrogate frame are then transported into the suspect's frame as $u_c^m = u_c\mathbf{X}_{\mathrm{align}}$.

The suspect's training constrains $\mathbf{W}_c$ to a shared subspace S spanned by node embeddings, so only the component of u_c within S contributes to detection (Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019). We extract S via singular value decomposition of $\mathbf{X}_{\mathrm{align}}$ because the top singular directions of $\mathbf{X}_{\mathrm{align}}$ span where the surrogate and suspect frames agree across architectures.

$$\mathbf{X}_{\mathrm{align}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

where $\sigma_1 \geq \sigma_2 \geq \cdots$ are the singular values on the diagonal of $\mathbf{\Sigma}$. The effective rank $k_{\mathrm{eff}} := |\{j : \sigma_j > \mathrm{tol}\,\sigma_1\}|$ is determined by a cutoff $\mathrm{tol} \in (0, 1)$. The shared subspace is

$$S := \mathrm{span}(\mathbf{V}_k) \subset \mathbb{R}^{d_{\mathbf{H}}},$$

where $\mathbf{V}_k$ contains the top k_{eff} right singular vectors, and S is formally defined in Appendix D.3. To avoid bias from unequal singular values, we calibrate with the whitening matrix $\mathbf{M} = \mathbf{V}_k \mathbf{\Sigma}_k^{-1} \mathbf{V}_k^\top$, where $\mathbf{\Sigma}_k$ are the top k_{eff} singular values of $\mathbf{X}_{\text{align}}$; $\mathbf{M}$ projects onto S and cancels per-direction amplification simultaneously, giving every direction in S uniform weight (full derivation in Appendix D.4). The calibrated basis $(\mathbf{V}_k, \mathbf{\Sigma}_k)$ then supports the detection statistic.

Provable Detection Test. Detection must yield more than an alignment score: radioactive watermark evidence requires a formal hypothesis test with a bounded false-positive rate τ, ensuring validity for any benign suspect regardless of architecture or training trajectory. We therefore cast detection as a statistical test on the per-class alignment between the owner's private direction u_c, defined in the surrogate space, and the suspect's classifier weight $\mathbf{W}_c$ inside the shared subspace S, converting alignment scores into p-values under an explicit null distribution.

This null distribution arises from a structural property of the keys: each u_c is sampled independently and never released, so under the benign null hypothesis $\mathcal{H}_0$ that the suspect was trained on benign data, u_c and $\mathbf{W}_c$ are statistically independent. Under the alternative $\mathcal{H}_1$ that the suspect was trained on watermarked data, $\mathbf{W}_c$ aligns with u_c due to the injected embedding shifts (Sablayrolles et al., 2020). The independence result follows the incomplete Beta distribution of two random unit vectors (Mardia & Jupp, 2009; Iscen et al., 2018), restated as Proposition 1 (Appendix D.1).

We formalize the alignment score as

$$s_c := \cos(\tilde{\mathbf{W}}_c, \tilde{u}_c),$$

where $\tilde{\mathbf{W}}_c = \mathbf{W}_c\mathbf{M}/\|\mathbf{W}_c\mathbf{M}\|_2$ and $\tilde{u}_c = u_c^m\mathbf{M}/\|u_c^m\mathbf{M}\|_2$ are whitened and normalized into S via matrix $\mathbf{M}$. Each s_c is converted into a per-class p-value p_c, and Fisher-combined (Fisher, 1925):

$$T = -2\sum_{c=1}^{C} \log p_c \sim \chi^2_{2C},$$

yielding the joint p-value p_{Fisher}. We reject $\mathcal{H}_0$ when $p_{\text{Fisher}} < \tau$ (e.g., $\tau = 0.05$). In practice, detection operates in two settings: **White-box**. Directly extract classifier weights $\mathbf{W}_c$ from f_θ; **Black-box**. Distill a student $\tilde{f}$ from f_θ's outputs and apply the same pipeline (Appendix F).

Detection Guarantee. Because suspects do not exist at embedding time, the owner cannot empirically test whether the watermark design will succeed once the dataset is released. Detection therefore requires an *advance guarantee*: a criterion that certifies, from design parameters alone, that detection will succeed with high probability against a future suspect. Corollary 1 provides this guarantee by giving a closed-form estimator of the Fisher-combined p-value p_{Fisher} in terms of design parameters.

Corollary 1 (Sufficient Condition for Detection Success)**.** *Let C be the number of classes, k_{eff} the effective rank of the shared subspace S ($3 \le k_{\text{eff}} \le C$), $\mu := \frac{1}{C}\sum_c s_c$ the average per-class alignment, τ the desired significance level, B the buyer pool size, $\delta_B \in (0,1)$ the Bernstein failure probability,* $\text{margin}(\delta_B)$ *the concentration margin at level δ_B, and $\mathcal{H}_0$ the null hypothesis that the suspect was trained on benign data. If*

$$\frac{C\, k_{\text{eff}}\, \mu^2}{2} - \text{margin}(\delta_B) \;>\; \ln(B/\tau), \tag{2}$$

then $\mathcal{H}_0$ is rejected at the Bonferroni-corrected level τ/B with probability $\ge 1 - \delta_B$.

This condition provides a $(1-\delta_B)$-probability guarantee of detection success before release: the LHS combines the design-time watermark strength proxy $C\, k_{\text{eff}}\, \mu^2/2$ with the concentration margin $\text{margin}(\delta_B)$, both computable at design time, while the right-hand side $\ln(B/\tau)$ is the rejection threshold for B buyers. The asymmetric scaling reflects the watermark's *radioactive* persistence and *scalable* attribution. The proof appears in Appendix D.5.

Among the parameters, C, τ, and B are fixed by the application. The owner controls two knobs: the effective rank k_{eff}, chosen at detection time as the dimension of S, and the per-class alignment μ, raised at injection time by the alignment target θ, perturbation bound δ, and watermark node count $|V_w|$. By tuning these knobs until Eq. 2 holds, the owner certifies detection success at $(1-\delta_B)$-probability in advance. Empirical ablations of these knobs are reported in Appendix G.2. Across all five datasets, the design-time estimator $C\, k_{\text{eff}}\, \mu^2/2$ tracks the realised $-\ln p_{\text{Fisher}}$ statistic closely, with tuned $(|V_w|, \theta, \delta)$ satisfying the condition on the four non-Amazon datasets, while Amazon-Ratings' small $C = 5$ leaves the concentration margin loose, as listed in Appendix D.5, Table 8.

3.4 SCALABLE OWNER ATTRIBUTION

Watermark detection answers whether a suspect is watermarked, but not which buyer leaked it. *Owner attribution* closes this gap by applying the alignment test to each of the B buyer keys and attributing the leak to the key with the strongest alignment. Because each buyer's private key u_c is an independent random unit vector within the shared subspace S, different buyers' keys remain mutually distinguishable, making attribution *scalable* as B grows.

Given a leaked suspect f_θ from buyer $b_{i^\star}$, the owner evaluates all B keys and assigns the leak to the buyer whose key yields the smallest Fisher p-value (Appendix E.1, Algorithm 3). Attribution succeeds when this buyer is $b_{i^\star}$, measured as the Top-1 accuracy.

Corollary 2 provides a closed-form approximation of the joint probability of detection and correct attribution. Success requires two conditions: (i) the true key $\mathcal{K}_{i^\star}$ must fire detection, and (ii) it must beat each of the $B-1$ foreign keys $\mathcal{K}_j$ ($j \neq i^\star$) in p-value. Each pairwise win depends only on the angle between the true and foreign direction vectors, which share a common spherical distribution.

Corollary 2 (Approximate Attribution Success Rate)**.** *Let $\varphi := \angle(u_c^{(\mathrm{own})}, u_c^{(\mathrm{foreign})})$ be the own–foreign angle, $T^\star := -\ln p_{\mathrm{Fisher}}^{(\mathrm{own})}$ the realized detection strength of the true key, and $f(\varphi) \propto \sin^{k_{\mathrm{eff}}-2}\varphi$ the density of φ on $[0, \pi/2]$. Define the foreign win probability as $P^\star(\varphi) := \Pr\big(p_{\mathrm{Fisher}}^{(\mathrm{foreign})} < p_{\mathrm{Fisher}}^{(\mathrm{own})} \mid \varphi\big)$ and its expectation $\bar{P}^\star := \int_0^{\pi/2} \exp(-T^\star \sin^2\varphi)\, f(\varphi)\, \mathrm{d}\varphi$. Then the Top-1 attribution success probability approximately satisfies*

$$\Pr\big(\textit{detection \& attribution success}\big) \;\approx\; \mathbf{1}[\mathbf{D}] \cdot (1 - \bar{P}^\star)^{B-1}, \tag{3}$$

where $\mathbf{D}$ is the event that the true key fires detection, ensured by Corollary 1's sufficient condition. The indicator $\mathbf{1}[\mathbf{D}]$ enforces own-key detection, and the exponent $B-1$ counts the foreign competitors. The proof appears in Appendix D.7.

When Corollary 1 holds, the approximation is tight; it saturates as the buyer pool B grows, supporting scalable attribution. The approximation directly yields the maximum sustainable scale of the buyer pool in practice. Empirical verification appears in Appendix D.7, Table 9.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We conduct experiments on five node classification graphs (descriptions and statistics are provided in Appendix C.3, Table 4): including three homophilic benchmarks (i.e., **ogbn-arxiv** (Hu et al., 2020), **ogbn-products** (Hu et al., 2020) and **Reddit** (Hamilton et al., 2017)) and two heterophilic benchmarks (i.e., **Amazon-Ratings** (Pei et al., 2020) and **Roman-Empire** (Pei et al., 2020)), spanning 22K–2.45M nodes, 100–602 features, 5–47 classes, and 0.05–0.81 homophily.

GNNs. We evaluate cross-architecture transfer on eight suspect architectures: **GCN** (Kipf & Welling, 2017) as the surrogate-matched reference; **GAT** (Velickovic et al., 2018) for attention-based message passing; **Graph Transformer** (Shi et al., 2021) for transformer-style global self-attention; **GraphSAGE** (Hamilton et al., 2017) for sampling-based inductive aggregation; **GIN** (Xu et al., 2019) for WL-powerful aggregation; **GCNII** (Chen et al., 2020) and **ResGCN** (Li et al., 2019) for deep residual architectures; and **APPNP** (Klicpera et al., 2019) for decoupled propagation. These models span attention-, sampling-, and propagation-based designs, shallow and deep architectures, as well as coupled and decoupled propagation schemes.

Baselines. Multi-buyer data-IP for node classification is a new threat model; no existing method transfers directly, so we adapt baselines from three related categories: (i) **RandPerturb**, the ablation of Ours, replaces the private-key-derived feature perturbation with random noise of the same perturbation budget. (ii) The next three are the in-domain graph watermarking methods. **Backdoor** (Xu et al., 2023), originally model-IP, we adapt to multi-buyer data-IP by assigning each buyer a distinct trigger. **BkdOwn** (Li et al., 2023), originally data-IP for graph classification, we adapt to node classification by treating each watermark node's k-hop subgraph as an individual graph and assigning each buyer a distinct clique. **ImpWM** (Zhang et al., 2024a), originally model-IP for graph node classification, we

adapt to data-IP by replacing the inner-loop suspect retraining, which the owner cannot control, with a fixed surrogate GNN. (iii) **Data Taggants** (Bouaziz et al., 2025), the cross-modal data-IP precedent, originally for image classification, we adapt to graph node classification by perturbing features and assigning each buyer an independent private key set. Full protocols appear in Appendix C.3.

Other methods are excluded under our setting, categorized in Appendix B, Table 2. Model-level methods plant the watermark in the owner's training pipeline. Model-fingerprinting methods extract a fingerprint post-training from the trained model. Neither modifies the released data, so no signal propagates through buyer training to the suspect model the owner verifies. Data-level methods verify on the released data, outside our threat model, which grants verifier access only to the suspect model.

Setup. We report four metrics under a unified *fidelity budget*. **Watermark Detection Accuracy (WDA)**: fraction of suspects correctly classified as watermarked or benign; **Owner Attribution Accuracy (OAA)**: fraction of watermarked suspects whose true buyer ranks top-1 among B keys; **Harmfulness**: accuracy loss on watermark nodes; and **Stealthiness**: imperceptibility of feature perturbations. Formal definitions are given in Appendix C.2. The fidelity budget constrains the accuracy drop on benign test nodes to $|\Delta\mathrm{Acc}| \leq 1\%$ for every dataset, ensuring tightly clustered accuracies across methods. As a result, cross-method differences reflect watermark quality rather than fidelity-strength trade-offs. Appendix C.3 details hyperparameters and implementation settings.

4.2 Owner Attribution at Multiple Buyers

Top-1 owner attribution at scale. We evaluate at marketplace scale with $B = 1000$ buyers per method, stress-testing attribution well beyond the few-buyer regime of prior watermarking work and matching the scale of contemporary data marketplaces. For each method, we train 50 watermarked suspects, each linked to a distinct owner probe sampled from the pool of 1000 probes, and 50 benign suspects. All methods are constrained to the fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$, where $\Delta\mathrm{Acc}$ denotes the accuracy drop on benign test nodes. Each of the 1000 probes is tested against each of the 100 suspects, yielding 1000×100 probe-suspect scores. A suspect is classified as watermarked if any probe exceeds the detection threshold, and the owner is attributed to the highest-scoring probe.

Table 1 shows that **Ours-WB (white-box setting)** achieves excellent *Watermark Detection Accuracy (WDA)* and *Owner Attribution Accuracy (OAA)* on four of the five datasets. On ogbn-arxiv, ogbn-products, Reddit, and Roman-Empire, WDA and OAA are 100% for Ours-WB; **Ours-BB (black-box setting)** attains OAA in $[94, 100]\%$ and WDA in $[92, 98]\%$. On these datasets, the tuned knobs $(|V_w|, \theta, \delta)$ satisfy Corollary 1, so the observed marketplace WDA and OAA are the empirical realization of the design-time detection guarantee. The empirical OAA values also agree with the closed-form estimate of per-buyer attribution probability at $B = 1000$ from Corollary 2, which predicts near-perfect attribution (Appendix D.7, Table 9). Beyond WDA and OAA, Ours retains competitive harmfulness and solid stealthiness across all datasets, confirming that the carrier-direction signal embeds without sacrificing model utility or watermark imperceptibility.

Amazon-Ratings is the only exception, with Ours-WB at $66/30$ and Ours-BB at $77/56$. Here the class count $C = 5$ combined with the 1% fidelity budget prevents Corollary 1 from holding. Amazon-Ratings is the only benchmark with such a small class count, whereas the other four span $C \in [18, 47]$. Through the upper bound $k_{\mathrm{eff}} \leq C$ from Assumption 1 in Appendix D.6, this directly caps k_{eff} at 5; the budget-allowed $|V_w| = 625$ only raises μ to ≈ 0.85, below the threshold required by Corollary 1 at this k_{eff}. Satisfying the condition even weakly would require, for example, increasing $|V_w|$ to ≈ 5000, which reaches only 2% Bernstein confidence (Appendix D.5, Table 8) while violating the fidelity budget, with the full trade-off shown in Appendix H.2, Figure 7; nevertheless the empirical OAA reaches 96%, which reflects the sufficient rather than necessary nature of Corollary 1.

Of the five comparison baselines, we run *Backdoor*, *RandPerturb*, and *Data Taggants* at the $B = 1000$ marketplace scale. *Data Taggants* is the strongest but still stays below Ours on every dataset, holding 90% OAA on ogbn-arxiv yet collapsing to 2% on Reddit. *Backdoor*'s cross-trigger activations cap WDA at 50–82% and collapse OAA to 0–20%, while *RandPerturb* yields WDA $\approx 50\%$ and OAA $\approx 0\%$ in both WB and BB. These results confirm that Ours' near-perfect WDA and OAA stem from the carrier-direction signal rather than arbitrary perturbations. We exclude *BkdOwn* and *ImpWM* because they fail before attribution. *ImpWM* misclassifies nearly all benign suspects as watermarked

Table 1: Marketplace owner attribution at $B = 1000$ (50 wm and 50 benign 2-layer GCN suspects) under fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$ per cell. WDA/OAA: Watermark Detection Accuracy / Owner Attribution Accuracy (%); $H, \hat{H}$: misclassification / watermark-induced change on V_w; $\mathcal{S}, \bar{\mathcal{S}}, \mathcal{S}_{\mathrm{knn}}$: max per-feature / mean Euclidean / k-NN distance perturbation on V_w; WB/BB: white/black-box access. Mean$_{\pm\mathrm{std}}$ over 50 wm suspects.

		Detection & Attribution		**Harmfulness**		**Stealthiness**		
Dataset	**Method**	WDA (%, ↑)	OAA (%, ↑)	H (↓)	$\hat{H}$ (↓)	$\mathcal{S}$ (↓)	$\bar{\mathcal{S}}$ (↓)	$\mathcal{S}_{\mathrm{knn}}$ (↓)
ogbn-arxiv	RandPerturb-WB	50.0	0.0	$0.59_{\pm 0.01}$	$+0.04_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$3.26_{\pm 0.00}$	$3.24_{\pm 0.00}$
	RandPerturb-BB	49.0	0.0	$0.59_{\pm 0.01}$	$+0.04_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$3.26_{\pm 0.00}$	$3.24_{\pm 0.00}$
	Backdoor (Xu et al., 2023)	50.0	18.0	$0.68_{\pm 0.05}$	$+0.37_{\pm 0.04}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	95.0	90.0	$\mathbf{0.17}_{\pm \mathbf{0.05}}$	$\mathbf{-0.36}_{\pm \mathbf{0.06}}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**100.0**	**100.0**	$0.38_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.55}_{\pm \mathbf{0.03}}$	$\mathbf{1.81}_{\pm \mathbf{0.03}}$
	Ours-BB	94.0	**100.0**	$0.38_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.55}_{\pm \mathbf{0.03}}$	$\mathbf{1.81}_{\pm \mathbf{0.03}}$
ogbn-products	RandPerturb-WB	51.0	0.0	$0.39_{\pm 0.00}$	$+0.00_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$2.88_{\pm 0.00}$	$5.32_{\pm 0.03}$
	RandPerturb-BB	49.0	0.0	$0.39_{\pm 0.00}$	$+0.00_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$2.88_{\pm 0.00}$	$5.32_{\pm 0.03}$
	Backdoor (Xu et al., 2023)	50.0	4.0	$\mathbf{0.28}_{\pm \mathbf{0.01}}$	$+0.01_{\pm 0.01}$	$50.64_{\pm 11.81}$	$4.82_{\pm 0.60}$	$5.77_{\pm 0.46}$
	Data Taggants (Bouaziz et al., 2025)	64.0	20.0	$0.34_{\pm 0.10}$	$\mathbf{-0.02}_{\pm \mathbf{0.01}}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$4.91_{\pm 0.03}$	$6.48_{\pm 0.15}$
	Ours-WB	**100.0**	**100.0**	$0.38_{\pm 0.01}$	$-0.01_{\pm 0.00}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.24}_{\pm \mathbf{0.05}}$	$\mathbf{4.52}_{\pm \mathbf{0.04}}$
	Ours-BB	92.0	94.0	$0.38_{\pm 0.01}$	$-0.01_{\pm 0.00}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.24}_{\pm \mathbf{0.05}}$	$\mathbf{4.52}_{\pm \mathbf{0.04}}$
Reddit	RandPerturb-WB	48.0	0.0	$0.08_{\pm 0.05}$	$+0.01_{\pm 0.05}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$7.08_{\pm 0.00}$	$19.18_{\pm 0.05}$
	RandPerturb-BB	47.0	0.0	$0.08_{\pm 0.05}$	$+0.01_{\pm 0.05}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$7.08_{\pm 0.00}$	$19.18_{\pm 0.05}$
	Backdoor (Xu et al., 2023)	82.0	10.0	$0.07_{\pm 0.00}$	$+0.02_{\pm 0.00}$	$10.86_{\pm 2.36}$	$5.34_{\pm 0.52}$	$18.08_{\pm 0.22}$
	Data Taggants (Bouaziz et al., 2025)	51.0	2.0	$\mathbf{0.06}_{\pm \mathbf{0.02}}$	$\mathbf{-0.01}_{\pm \mathbf{0.01}}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$9.08_{\pm 0.75}$	$20.16_{\pm 0.68}$
	Ours-WB	**100.0**	**100.0**	$0.07_{\pm 0.01}$	$-0.00_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.39}_{\pm \mathbf{0.05}}$	$\mathbf{17.77}_{\pm \mathbf{0.05}}$
	Ours-BB	98.0	**100.0**	$0.07_{\pm 0.01}$	$-0.00_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.39}_{\pm \mathbf{0.05}}$	$\mathbf{17.77}_{\pm \mathbf{0.05}}$
Amazon-Ratings	RandPerturb-WB	52.0	0.0	$0.40_{\pm 0.01}$	$-0.19_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.00_{\pm 0.00}$	$4.95_{\pm 0.00}$
	RandPerturb-BB	49.0	0.0	$0.40_{\pm 0.01}$	$-0.19_{\pm 0.01}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.00_{\pm 0.00}$	$4.95_{\pm 0.00}$
	Backdoor (Xu et al., 2023)	50.0	0.0	$0.66_{\pm 0.04}$	$+0.24_{\pm 0.04}$	$1.09_{\pm 0.18}$	$3.50_{\pm 0.81}$	$3.48_{\pm 0.78}$
	Data Taggants (Bouaziz et al., 2025)	70.0	40.0	$0.33_{\pm 0.05}$	$-0.27_{\pm 0.09}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$7.68_{\pm 0.25}$	$7.53_{\pm 0.25}$
	Ours-WB	66.0	30.0	$\mathbf{0.28}_{\pm \mathbf{0.04}}$	$\mathbf{-0.31}_{\pm \mathbf{0.04}}$	$0.60_{\pm 0.00}$	$\mathbf{1.20}_{\pm \mathbf{0.12}}$	$\mathbf{1.29}_{\pm \mathbf{0.11}}$
	Ours-BB	**77.0**	**56.0**	$\mathbf{0.28}_{\pm \mathbf{0.04}}$	$\mathbf{-0.31}_{\pm \mathbf{0.04}}$	$0.60_{\pm 0.00}$	$\mathbf{1.20}_{\pm \mathbf{0.12}}$	$\mathbf{1.29}_{\pm \mathbf{0.11}}$
Roman-Empire	RandPerturb-WB	52.0	0.0	$0.45_{\pm 0.01}$	$+0.05_{\pm 0.02}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.00_{\pm 0.00}$	$4.98_{\pm 0.00}$
	RandPerturb-BB	49.0	0.0	$0.45_{\pm 0.01}$	$+0.05_{\pm 0.02}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.00_{\pm 0.00}$	$4.98_{\pm 0.00}$
	Backdoor (Xu et al., 2023)	50.0	20.0	$0.92_{\pm 0.02}$	$+0.57_{\pm 0.03}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.78}$
	Data Taggants (Bouaziz et al., 2025)	80.0	60.0	$0.33_{\pm 0.05}$	$-0.07_{\pm 0.08}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	**100.0**	**100.0**	$\mathbf{0.16}_{\pm \mathbf{0.01}}$	$\mathbf{-0.25}_{\pm \mathbf{0.02}}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.67}_{\pm \mathbf{0.09}}$	$\mathbf{1.73}_{\pm \mathbf{0.09}}$
	Ours-BB	98.0	**100.0**	$\mathbf{0.16}_{\pm \mathbf{0.01}}$	$\mathbf{-0.25}_{\pm \mathbf{0.02}}$	$\mathbf{0.50}_{\pm \mathbf{0.00}}$	$\mathbf{1.67}_{\pm \mathbf{0.09}}$	$\mathbf{1.73}_{\pm \mathbf{0.09}}$

even at single-buyer scale (Appendix G.1, Table 16), while *BkdOwn*'s owner and foreign probes cross-fire indistinguishably at $B = 2$ (Appendix H.5, Table 25), so scaling to $B = 1000$ is infeasible.

Appendix H.1 extends the cross-architecture analysis to seven additional suspects (GAT-2L, Graph Transformer-2L, GraphSAGE-2L, GIN-2L, GCNII-8L, ResGCN-8L, APPNP-2L); Ours-WB/BB attains $\geq 90\%$ OAA throughout, while *Backdoor* falls to chance and *Data Taggants* swings 2–90%.

4.3 Robustness and Scalability of Owner Attribution

We evaluate watermark survival against random perturbation, model pruning, fine-tuning, and knowledge distillation attacks at $B = 1000$; Ours remains robust across all, while *Backdoor* never exceeds 17% OAA and *Data Taggants* collapses under distillation (Appendix H.3). We also test attribution scalability across $B \in [1, 1000]$; Ours sustains OAA $\approx 100\%$ across all B on four datasets, while *Backdoor* drops to $\leq 20\%$ and even *Data Taggants* collapses from 90% to 2% (Appendix H.4).

5 Conclusions

We propose the first **radioactive** graph node classification dataset watermark for **scalable** owner attribution with **architecture-agnostic** verification, backed by a closed-form high-probability detection guarantee, verified on five benchmarks and eight GNN architectures with reliable cross-architecture detection and per-buyer attribution among 1000 buyers at a strict 1% fidelity budget.

Our study has three limitations. (i) The detection guarantee assumes at least three classes, with attribution degrading at small class counts under strict fidelity budgets. (ii) Empirical validation is restricted to node classification; extension to link prediction, graph-level classification, and inductive or temporal settings remains open. (iii) Adaptive adversaries that partially reconstruct the surrogate to launch gradient-based removal attacks are not characterised. We discuss cross-modal extensions, a lighter black-box variant, and stronger adaptive-adversary characterisation in Appendix A.

REFERENCES

Amazon Web Services. AWS Data Exchange. https://aws.amazon.com/data-exchange/, 2026. Accessed 2026-09.

Venkata Sai Pranav Bachina, Aaryan Ajay Sharma, Charu Sharma, and Ankit Gangwal. GENIE: watermarking graph neural networks for link prediction. *Transactions on Machine Learning Research*, 2026.

Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, 1998.

Wassim Bouaziz, Nicolas Usunier, and El-Mahdi El-Mhamdi. Data taggants: Dataset ownership verification via harmless targeted data poisoning. In *International Conference on Learning Representations*, 2025.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pp. 1725–1735, 2020.

Aloni Cohen, Alexander Hoover, and Gabe Schoenbach. Watermarking language models for many adaptive users. In *Symposium on Security and Privacy*, pp. 2583–2601, 2025.

Enyan Dai, Minhua Lin, Xiang Zhang, and Suhang Wang. Unnoticeable backdoor attacks on graph neural networks. In *World Wide Web Conference*, pp. 2263–2273, 2023.

Enyan Dai, Minhua Lin, and Suhang Wang. Pregip: Watermarking the pretraining of graph neural networks for deep IP protection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 415–426, 2025.

Jane Downer, Ren Wang, and Binghui Wang. Watermarking graph neural networks via explanations for ownership protection. *arXiv preprint arXiv:2501.05614*, 2025.

Olive Jean Dunn. Multiple comparisons among means. *Journal of the American Statistical Association*, 56(293):52–64, 1961.

Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 1925.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7 (2):179–188, 1936.

Roberto Gheda, Abele Mălan, Robert Birke, Maksim Kitsak, and Lydia Chen. Checkmate! watermarking graph diffusion models in polynomial time. In *International Conference on Learning Representations*, pp. 117330–117352, 2026.

Fei Guo, Jianmin Wang, and Deyi Li. Fingerprinting relational databases. In *Symposium on Applied Computing*, pp. 487–492, 2006.

Junfeng Guo, Yiming Li, Lixu Wang, Shu-Tao Xia, Heng Huang, Cong Liu, and Bo Li. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. In *Advances in Neural Information Processing Systems*, pp. 54421–54450, 2023.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1024–1034, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, pp. 22118–22133, 2020.

Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael G. Rabbat, and Hervé Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Transactions on Big Data*, 4(1):65–77, 2018.

Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. CRC Press, 2014.

Nicolas Keriven and Gabriel Peyré. Universal invariant and equivariant graph neural networks. In *Advances in Neural Information Processing Systems*, pp. 7090–7099, 2019.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Guohao Li, Matthias Müller, Ali K. Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *International Conference on Computer Vision*, pp. 9266–9275, 2019.

Jiasen Li, Yanwei Liu, Zhuoyi Shang, Xiaoyan Gu, and Weiping Wang. DRGW: learning disentangled representations for robust graph watermarking. In *World Wide Web Conference*, pp. 3263–3274, 2026.

Jipeng Li and Yannning Shen. Robust GNN watermarking via implicit perception of topological invariants. *arXiv preprint arXiv:2510.25934*, 2025.

Tingzhi Li and Xuefeng Liu. Wgle:backdoor-free and multi-bit black-box watermarking for graph neural networks. *arXiv preprint arXiv:2506.08602*, 2025.

Yiming Li, Mingyan Zhu, Xue Yang, Yong Jiang, Tao Wei, and Shu-Tao Xia. Black-box dataset ownership verification via backdoor watermarking. *IEEE Transactions on Information Forensics and Security*, 18:2318–2332, 2023.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample-variance penalization. In Adam Tauman Kalai and Mehryar Mohri (eds.), *Conference on Learning Theory*, 2009.

Ari S. Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. In *Advances in Neural Information Processing Systems*, pp. 5732–5741, 2018.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Hongrui Peng, Haolang Lu, Yuanlong Yu, Weiye Fu, Kun Wang, and Guoshun Nan. Kgmark: A diffusion watermark for knowledge graphs. In *International Conference on Machine Learning*, 2025.

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pp. 6076–6085, 2017.

Saksham Rastogi, Pratyush Maini, and Danish Pruthi. STAMP your content: Proving dataset membership via watermarked rephrasings. In *International Conference on Machine Learning*, 2025.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning*, pp. 8326–8335, 2020.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *Advances in Neural Information Processing Systems*, 2024.

Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *International Joint Conference on Artificial Intelligence*, pp. 1548–1554, 2021.

Snowflake Inc. Snowflake marketplace. `https://www.snowflake.com/en/data-cloud/marketplace/`, 2026. Accessed 2026-09.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2026.

Asim Waheed, Vasisht Duddu, and N. Asokan. Grove: Ownership verification of graph neural networks using embeddings. In *IEEE Symposium on Security and Privacy*, pp. 2460–2477, 2024.

Haiming Wang, Zhikun Zhang, Min Chen, and Shibo He. Making watermark survive model extraction attacks in graph neural networks. In *International Conference on Communications*, pp. 57–62, 2023.

Biao Wu, Qiang Huang, and Anthony K. H. Tung. From zero to hero: Detecting leaked data through synthetic data injection and model querying. *Proceedings of the VLDB Endowment*, 17(8): 1898–1910, 2024.

Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *USENIX Security Symposium*, pp. 1523–1540, 2021.

Jing Xu, Stefanos Koffas, Oguzhan Ersoy, and Stjepan Picek. Watermarking graph neural networks based on backdoor attacks. In *European Symposium on Security and Privacy*, pp. 1179–1197, 2023.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Yuxin Yang, Qiang Li, Yuan Hong, and Binghui Wang. Fedgmark: Certifiably robust watermarking for federated graph learning. In *Advances in Neural Information Processing Systems*, 2024.

Xiaoyu You, Youhe Jiang, Jianwei Xu, Mi Zhang, and Min Yang. Gnnfingers: A fingerprinting framework for verifying ownerships of graph neural networks. In *World Wide Web Conference*, pp. 652–663, 2024.

Linji Zhang, Mingfu Xue, Leo Yu Zhang, Yushu Zhang, and Weiqiang Liu. An imperceptible and owner-unique watermarking method for graph neural networks. In *ACM Turing Award Celebration Conference*, 2024a.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *ACM Symposium on Access Control Models and Technologies*, pp. 15–26, 2021.

Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4386–4397, 2024b.

Xiangyu Zhao, Hanzhou Wu, and Xinpeng Zhang. Watermarking graph neural networks by random graphs. In *International Symposium on Digital Forensics and Security*, pp. 1–6, 2021.

Yihao Zheng, Jinfei Liu, Kui Ren, and Li Xiong. B^2mark: A blind and buyer-traceable watermarking scheme for tabular datasets. *Proceedings of the ACM on Management of Data*, 3(4):1–26, 2025.

Allan Zhou, Kaien Yang, Kaylee Burns, Adriano Cardace, Yiding Jiang, Samuel Sokota, J. Zico Kolter, and Chelsea Finn. Permutation equivariant neural functionals. In *Advances in Neural Information Processing Systems*, 2023.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations*, 2019.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2847–2856, 2018.

AI USE STATEMENT

In this work, we have not used generative AI tools for theoretical model development, mathematical formulation, proofs, hypothesis refinement, research methodology design, implementation, or result interpretation. Synthetic data generation, translation, data cleaning, and qualitative analysis are not applicable to this work.

ETHICS STATEMENT

This work develops a defensive intellectual property protection technique for graph datasets released under restricted-use agreements. All experiments use publicly available benchmark datasets; no new data were collected, no human subjects were involved, and no personally identifiable information is used. The intended use is to help legitimate dataset owners verify ownership of leaked data and trace it to the specific buyer responsible, providing evidence to enforce license terms.

REPRODUCIBILITY STATEMENT

We provide the full implementation of our models and experimental setup to ensure reproducibility. Experimental results are reported as mean and standard deviation across different random seeds. Hyperparameters, training protocols, and baseline configurations are detailed in Appendix C.3. **Our code and data are available at `https://anonymous.4open.science/r/graph_watermark-0C2F/README.md`.**

A FUTURE WORK AND EXTENSIONS

Our core mechanism, embedding owner-specific carrier directions that survive in a shared subspace of any model trained on the data, is not GNN-specific; instantiating it on tabular data, vision transformers, or LLM fine-tuning IP would test whether owner-identifiable watermarking generalises across modalities. Extending the framework to other graph tasks such as link prediction and graph-level classification requires re-targeting the embedding objective and detection statistic. A lighter black-box variant could bind each private key to a target label and verify ownership from the suspect's loss or logits alone, removing the need for embedding access. Characterising the Pareto-optimal trade-off between watermark removal and the suspect's clean-task accuracy under stronger adaptive adversaries remains an open challenge.

B RELATED WORK

Beyond the *cross-fire* failure identified in §1, prior methods exhibit further design gaps for node classification: BkdOwn (Li et al., 2023) is designed for graph classification and Backdoor (Xu et al., 2023) originally targets model watermarking rather than datasets. Their trigger mechanisms also exhibit three issues: *preset-label triggers* based on label flipping degrade utility, *structural triggers* are amplified by message passing to create unresolvable stealth-faithfulness conflicts, and *fixed feature triggers*, adopted to replace structural triggers, leave detectable frequency anomalies (§H.3, Table 24). KGMark (Peng et al., 2025) uses a private key to perturb pre-trained knowledge graph (KG) embeddings in the frequency domain, but its signal does not survive downstream GNN retraining, confining applicability to KG link prediction. As a consequence, none of these methods can support scalable owner attribution across a full marketplace.

Existing GNN IP protection methods divide by the IP they protect: (i) *data-level watermarking* modifies the released graph, so any derived model inherits the signal, covering both dataset and model IP; (ii) *model-level watermarking* controls the training pipeline, covering only the model; (iii) *model fingerprinting* extracts a post-hoc provenance signature from model behavior, also covering only the model. Figure 3 contrasts data-level and model-level approaches, and model fingerprinting shares the same gap since its signal is never embedded in the released data. Table 2 enumerates concrete methods and their mechanisms.

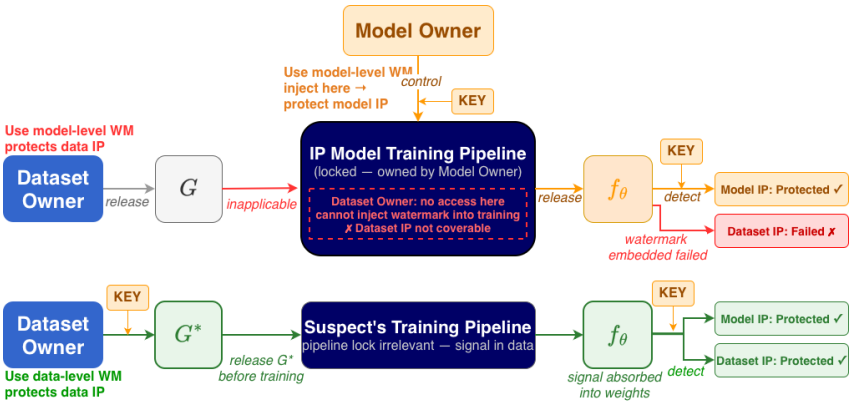


Figure 3: **Scheme contrast: model-level WM cannot protect dataset IP; data-level WM can.**

Data-level watermarking. Three prior methods modify the released graph or its embeddings to protect dataset ownership without controlling downstream training. One uses backdoor watermarking (BkdWM) via planted triggers, while the others use data-perturbation watermarking (DPWM) via trigger-free private-key signatures. **BkdOwn** (Li et al., 2023) injects a backdoor trigger subgraph into graph-classification data and verifies via a t-test on model outputs. Its triggers cross-fire across buyers at marketplace scale, making owner attribution unreliable, and it requires preset target labels and structural perturbation. **KGMark** (Peng et al., 2025) embeds a private-key signature into pre-trained knowledge graph (KG) entity embeddings and verifies by a gray-box likelihood-ratio test on the embedding residual. It assumes an embedding-redistribution threat in which the embedding artifact itself is leaked. The signal does not survive downstream model retraining and is restricted to KG link prediction. The single-owner design does not extend to per-buyer attribution. **DRGW** (Li et al., 2026) embeds a private-key Gaussian watermark into the released graph as discrete edge edits and verifies by a likelihood-ratio test on the suspect graph. The watermark does not survive downstream model training and is restricted to link prediction. The single-watermark verifier does not scale to multi-buyer attribution.

None of the three extends to multi-buyer attribution for node classification datasets. Among them, only BkdOwn verifies on a trained downstream model. Since the dataset is licensed not to be redistributed and remains outside the owner's reach, the trained model is our only verification target. We therefore adapt BkdOwn from graph classification to node classification as a baseline. KGMark and DRGW verify on the released embedding or graph directly, leaving the watermark-to-model relationship undefined. Adapting them to verify on the trained model would require designing this watermark-to-model relationship from scratch, amounting to a new method, so KGMark and DRGW cannot serve as baselines in our setting.

Model-level watermarking. Model-level methods actively embed the watermark by controlling the protected model's training pipeline, yielding model IP only. **Backdoor watermarking (BkdWM)** trains the model to flip the predicted label at trigger inputs to an owner-chosen target: RandGraph (Zhao et al., 2021) and ExtrWM (Wang et al., 2023) embed a backdoor by training the model to respond to random subgraph triggers, for node and graph classification respectively, without modifying the released graph. Backdoor (Xu et al., 2023) poisons the released graph with structural and feature triggers for graph and node classification. FedGMark (Yang et al., 2024) plants structural triggers in the released graph for federated graph classification. ImpWM (Zhang et al., 2024a) injects imperceptible trigger subgraphs at relabelled anchor nodes in the released graph for node classification. GENIE (Bachina et al., 2026) extends the same training-time backdoor mechanism to link prediction without modifying the released graph. Backdoor *attacks* on GNNs (Zhang et al., 2021; Xi et al., 2021; Dai et al., 2023; Zhang et al., 2024b) use a similar trigger mechanism but target malicious prediction manipulation, not IP protection, and fall outside our threat model. **Behavioral watermarking (BehWM)** trains the model to shape the output distribution at trigger inputs while preserving the predicted label. WGLE (Li & Liu, 2025) modulates local degree deviation entropy as a behavioral signal for node classification, verified by an empirical threshold rather than a formal hypothesis test. **Parameter watermarking (ParWM)** trains the model to encode the watermark in the statistical distribution of weights or internal representations, independent of any

input pattern: PreGIP (Dai et al., 2025) targets self-supervised pretrained graph encoders for graph classification. InvGNN-WM (Li & Shen, 2025) exploits graph spectral invariants for graph and node classification, verified by a Hoeffding-based test. WMExpl (Downer et al., 2025) targets the GNN explanation space for node classification, verified by a z-test with NP-hard removal guarantees. CheckWate (Gheda et al., 2026) embeds a private key into graph diffusion generative models, verified by a polynomial-time scheme.

None of the model-level methods extends to multi-buyer attribution, and the required training control is unavailable in our data-release setting. Among the three sub-mechanisms, only data-modifying BkdWM methods establish the watermark-to-model relationship needed for verification. Triggers planted in the released graph migrate through the buyer's training to embed as a backdoor in the trained model. The owner verifies by querying the trained model with a trigger and observing the planted target-label response. Within data-modifying BkdWM, FedGMark targets federated graph classification, mismatching our centralized node classification setting. We adapt Backdoor and ImpWM as baselines for their applicability to node classification and trigger designs that yield a distinct trigger per buyer for multi-buyer attribution. BehWM, ParWM, and non-data-modifying BkdWM methods leave the released data unmarked, so there is no data-side signal to propagate through training and be detected in the buyer's trained model. The only alternative is to directly detect a watermark already embedded in the model, but such a watermark would have to be planted by the trainer, who in our setting is the buyer suspected of unauthorized data use. The buyer will not plant a watermark proving their own violation, so these methods cannot establish dataset usage.

Model fingerprinting. Model fingerprinting passively extracts a behavioral fingerprint (BehFP) from the trained model. The owner records the model's responses on a probe set chosen post-training as a reference, and verifies by matching the suspect model's responses on the same probes. GrOVe (Waheed et al., 2024) checks embedding similarity to the owner's reference model via a learned classifier with gray-box access to intermediate embeddings for node classification. The authors show that independently trained models on the same dataset produce no detectable signal. GNNFingers (You et al., 2024) constructs adversarial fingerprint queries post-hoc to detect model derivation such as fine-tuning, pruning, or distillation rather than dataset lineage, using black-box access for graph and node classification. A suspect who independently trains on the owner's data yields a different fingerprint.

None of the model fingerprinting methods extends to multi-buyer attribution. BehFP follows the same failure pattern as Model-level methods. With no data signal to propagate through training, detection would have to fingerprint the buyer's model directly with a probe set and reference. But the buyer suspected of unauthorized data use will not provide a probe set and reference proving their own violation, so these methods cannot establish dataset usage either.

Cross-modal data-IP precedents. Data-IP watermarking has been studied beyond graphs across image, text, tabular and relational modalities. **Image**: Radioactive Data (Sablayrolles et al., 2020) plants single-buyer feature perturbations on image training data detected by a white-box statistical test on the trained model. Data Taggants (Bouaziz et al., 2025) extends Radioactive Data with harmless targeted poisoning that preserves model utility, mechanistically closest to our design via gradient matching on data perturbations with binomial detection. We adapt this mechanism to graph node features as a cross-modal baseline. Data Taggants itself has no mechanism that guarantees multi-buyer attribution. Boneh-Shaw (Boneh & Shaw, 1998) provides theoretical fingerprinting codes that mathematically identify at least one colluder when multiple buyers combine their fingerprinted copies, but does not specify how to embed the code into graph data, so the theory remains a reference rather than a runnable baseline. **Text and LLM**: Rastogi et al. (Rastogi et al., 2025) mark LLM training data via single-buyer text paraphrases that propagate to the trained language model, but paraphrasing rewrites text in different words while preserving meaning, which has no equivalent for graph structure, so it does not transfer to graph data-IP. Sander et al. (Sander et al., 2024) watermark LLM output at inference time, bypassing training data so no signal propagates to a buyer-trained model. Cohen et al. (Cohen et al., 2025) provide multi-buyer watermarks for LLMs similarly at inference time, so the multi-buyer scheme does not transfer to our data-release setting either. **Tabular and relational**: Wu et al. (Wu et al., 2024) inject synthetic clean-label rows into regions of the feature space that contain no original samples, labelled to shift the local class distribution, and detect prediction divergence on samples drawn from the same regions. A graph version must either isolate

Table 2: Comparison of watermarking methods in scope. Methods in **bold** are used as baselines in our experiments.

Method	Mech.	Key	Graph Pert.	Train Ctrl.	Task	Detection	IP
Data-level watermarking							
BkdOwn (Li et al., 2023)	BkdWM	Trigger	S	×	GC	BB	D+M
KGMark (Peng et al., 2025)	DPWM	Private k	F	×	KGE	GB	D
DRGW (Li et al., 2026)	DPWM	Private k	S	×	LP	GB	D
Ours	DPWM	Private k	F	×	NC	WB/BB	D+M
Model-level watermarking							
RandGraph (Zhao et al., 2021)	BkdWM	Trigger	–	✓	NC	BB	M
ExtrWM (Wang et al., 2023)	BkdWM	Trigger	–	✓	GC	BB	M
Backdoor (Xu et al., 2023)	BkdWM	Trigger	S,F	✓	GC/NC	BB	M
FedGMark (Yang et al., 2024)	BkdWM	Trigger	S	✓	GC	BB	M
ImpWM (Zhang et al., 2024a)	BkdWM	Trigger	S	✓	NC	BB	M
GENIE (Bachina et al., 2026)	BkdWM	Trigger	–	✓	LP	BB	M
WGLE (Li & Liu, 2025)	BehWM	Trigger	–	✓	NC	BB	M
PreGIP (Dai et al., 2025)	ParWM	Trigger	–	✓	GC	BB	M
InvGNN-WM (Li & Shen, 2025)	ParWM	Trigger	–	✓	GC/NC	BB	M
WMExpl (Downer et al., 2025)	ParWM	–	–	✓	NC	BB	M
CheckWate (Gheda et al., 2026)	ParWM	Private k	–	✓	GG	BB	M
Model fingerprinting							
GrOVe (Waheed et al., 2024)	BehFP	–	–	×	NC	GB	M
GNNFingers (You et al., 2024)	BehFP	Private k	–	×	GC/NC	BB	M
Cross-modal data-IP precedent							
Data Taggants (Bouaziz et al., 2025)	DPWM	Private k	F	×	IC	BB	D+M

the synthetic nodes, where their zero degree trivially exposes them, or connect them, where neighbor aggregation dissolves the local class shift, so we do not adopt it as a cross-modal baseline. Guo et al. (Guo et al., 2006) and the recent (Zheng et al., 2025) develop multi-buyer fingerprinting codes for relational data verified on the leaked data copy itself, leaving the watermark-to-model relationship undefined and therefore unable to serve as a baseline.

Across modalities, Data Taggants is the only cross-modal precedent we adopt as a baseline. Its modality-agnostic mechanism of gradient matching with binomial detection transfers directly to graph node features.

Column legend for Table 2. **Mech.**: *DPWM*, trigger-free data-perturbation watermark on released data; *BkdWM*, watermark that flips the predicted label at trigger inputs to an owner-chosen target; *BehWM*, watermark that shapes the output distribution at trigger inputs while preserving the predicted label; *ParWM*, watermark encoded in the model's weights or internal representations, independent of input pattern; *BehFP*, fingerprint passively extracted from the trained model via a probe set and reference. **Key**: private artifact for verification; *Trigger* = specific input pattern, observable when used; *Private* k = cryptographic secret known only to the owner; – = no key. **Graph Pert.**: how the released data is modified; S = structure; F = node features; – = unmodified. **Train Ctrl.**: does the owner need to control the buyer's training? ✓ = yes; × = no. **Task**: NC = node classification, GC = graph classification, LP = link prediction, KGE = knowledge graph embedding, GG = graph generation, IC = image classification. **Detection**: *WB* = white-box (architecture + weights); *GB* = gray-box (intermediate embeddings); *BB* = black-box (predictions only). For data-level methods, GB indicates direct verifier access to the released data artifact rather than querying a derived model. **IP**: protection scope; *D+M* = dataset and model; *M* = model only; *D* = dataset only.

C Setup and Definitions

C.1 Notation Table

Table 3 lists the core notation used throughout the paper.

C.2 Evaluation Metrics

We report four primary metrics: *Watermark Detection Accuracy* (WDA), *Owner Attribution Accuracy* (OAA), *Harmfulness*, and *Stealthiness* (§C.2.1). All four are evaluated under a per-cell fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$ on benign test nodes (§C.2.2). §C.2.3 additionally defines the per-suspect primitives at $B = 1$ (SS, VSR, AUC, TPR/FPR) that feed WDA/OAA.

Table 3: Summary of core notation.

Symbol	Description	Symbol	Description
Graph and Data		*Watermark Detection*	
$G = (V, E, \mathbf{X})$	Graph: nodes, edges, node features	$\mathbf{X}_{\text{align}}$	Surrogate-to-suspect alignment matrix
$\mathcal{D}_{\text{wm}}$	Watermarked dataset	k_{eff}	Effective rank of $\mathbf{X}_{\text{align}}$
$V_w \subset V$	Watermark node subset	S	Shared embedding subspace
C	Number of classes	$\mathbf{M}$	Whitening matrix, calibrates alignment into S
Watermark Mechanism		s_c	Whitened cosine alignment for class c
u_c	Per-class private key (random unit vec.)	$\mu = \frac{1}{C}\sum_c s_c$	Mean per-class alignment ($\mathcal{H}_1$ signal)
$\Delta\mathbf{x}_v$	Owner's feature perturbation, $v \in V_w$	p_{Fisher}	Fisher-combined p-value over C classes
$\Delta\mathbf{H}_v$	Embedding shift on $v \in V_w$ toward u_c	$\kappa = \log_{10}(\tau/B)$	Bonferroni-corrected $\log p$ threshold
θ	Alignment target (cos stop)	δ_B	Bernstein failure probability
δ	Feature perturbation bound	*Marketplace and Metrics*	
GNN Models		B	Number of buyers
f_θ	Suspect GNN trained on $\mathcal{D}_{\text{wm}}$	τ	Per-comparison significance level
f_s	Owner's surrogate GNN	ΔAcc	Fidelity: accuracy change vs. benign
$\mathbf{W}_c$	Per-class linear classifier head weight of suspect		

C.2.1 Primary Metrics

(1) Watermark Detection Accuracy (WDA) measures whether the verifier correctly classifies each suspect as watermarked or benign. Given a pool of $n = n_+ + n_-$ suspects (n_+ watermarked, n_- benign) with true labels $z_j \in \{0, 1\}$ ($z_j = 1$ iff suspect j is watermarked), WDA is the accuracy of the per-suspect *fire* decision $\hat{z}_j \in \{0, 1\}$:

$$\text{WDA} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}[\hat{z}_j = z_j] \in [0, 1], \quad \text{higher is better,} \tag{4}$$

where $\mathbf{1}[\cdot]$ is the indicator function (1 if the condition holds, 0 otherwise). The *fire* decision uses "any-key fires": $\hat{z}_j = 1$ if any of the B probes fires on suspect j, and $\hat{z}_j = 0$ otherwise.

(2) Owner Attribution Accuracy (OAA) measures whether the verifier *both* detects and correctly attributes each watermarked suspect: attribution counts only when the suspect first fires ($\hat{z}_j = 1$) *and* the strongest probe matches the true owner. Each watermarked suspect j has a unique true owner $i_j^\star \in \{1, \ldots, B\}$; the verifier's predicted owner $\hat{i}_j \in \{1, \ldots, B\}$ is the probe yielding the strongest detection signal on suspect j. OAA is the joint success rate over the n_+ watermarked suspects:

$$\text{OAA} = \frac{1}{n_+}\sum_{j=1}^{n_+} \mathbf{1}\left[\hat{z}_j = 1 \text{ and } \hat{i}_j = i_j^\star\right] \in [0, 1], \quad \text{higher is better.} \tag{5}$$

(3) Harmfulness (Guo et al., 2023) measures whether the watermark degrades predictions on the watermark node set V_w under each model's deployment condition. We report two quantities: the absolute miss rate H of the watermarked model f^w, trained on $\mathcal{D}_{\text{wm}}$, on its deployment input $\mathbf{X}_{\text{wm}}$, and the baseline-corrected increment $\hat{H}$ that subtracts miss rate of the matched benign model f^b, trained on $\mathcal{D}$, on $\mathbf{X}_{\text{orig}}$ to isolate the watermark-specific harm beyond V_w nodes' natural classification difficulty; lower is preferable for both.

H is the fraction of V_w that f^w misclassifies on its deployment input:

$$H = \frac{1}{|V_w|}\sum_{v \in V_w} \mathbf{1}[f^w(\mathbf{A}, \mathbf{X}_{\text{wm}}, v) \neq y_v]. \tag{6}$$

V_w is method-specific: for Ours, an owner-chosen subset (§3.2); for trigger baselines, the method-native trigger node set (§C.2.3).

$\hat{H}$ subtracts f^b's baseline miss rate on the same nodes' original features:

$$\hat{H} = H - \frac{1}{|V_w|}\sum_{v \in V_w} \mathbf{1}[f^b(\mathbf{A}, \mathbf{X}, v) \neq y_v]. \tag{7}$$

(4) Stealthiness (Li et al., 2023) measures the input-feature imperceptibility of the watermark. We report three quantities: the worst-case ℓ_∞ (per-feature maximum) perturbation $\mathcal{S}$ and mean ℓ_2

(Euclidean) perturbation $\bar{\mathcal{S}}$ on V_w, both applicable when a per-node baseline $\mathbf{x}_{\mathrm{orig,v}}$ exists, and the distribution-level $\mathcal{S}_{\mathrm{knn}}$. Smaller is more imperceptible for all three.

$\mathcal{S}$ is the worst-case ℓ_∞ perturbation across V_w:

$$\mathcal{S} = \max_{v \in V_w} \|\Delta \mathbf{x}_v\|_\infty. \tag{8}$$

$\bar{\mathcal{S}}$ is the mean ℓ_2 perturbation:

$$\bar{\mathcal{S}} = \frac{1}{|V_w|} \sum_{v \in V_w} \|\Delta \mathbf{x}_v\|_2. \tag{9}$$

$\mathcal{S}_{\mathrm{knn}}$ is the mean k-nearest-neighbours (k-NN) distance ($k=5$) from each watermark-node feature $\mathbf{x}_{\mathrm{wm},v}$ ($v \in V_w$) to its k nearest neighbours among the non-watermark nodes' features $\{\mathbf{x}_{\mathrm{orig},u} : u \in V \setminus V_w\}$:

$$\mathcal{S}_{\mathrm{knn}} = \frac{1}{|V_w|} \sum_{v \in V_w} \frac{1}{k} \sum_{u \in k\text{-NN}(v)} \|\mathbf{x}_{\mathrm{wm},v} - \mathbf{x}_{\mathrm{orig},u}\|_2, \tag{10}$$

where k-NN(v) denotes the k non-watermark nodes whose features are closest to $\mathbf{x}_{\mathrm{wm},v}$ in Euclidean distance.

C.2.2 Fidelity Budget (Constraint, Not a Comparator)

All reported cells satisfy a per-cell fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$ on the Accuracy Gap $\Delta\mathrm{Acc}$ between watermarked and benign models, treated as a *constraint* that every reported cell must pass, not a comparison column. Cross-method differences on the four primary metrics (§C.2.1) thus reflect method quality at matched utility, ruling out asymmetric fidelity-strength trade-offs.

$\Delta\mathrm{Acc}$ is the accuracy difference between the watermarked suspect f^w and a matched benign baseline f^b on the held-out benign test set $\mathcal{V}_{\mathrm{test}}$, which isolates the training-time effect from any test-time perturbation:

$$\Delta\mathrm{Acc} = \mathrm{Acc}(f^w) - \mathrm{Acc}(f^b), \quad \mathrm{Acc}(f) = \frac{1}{|\mathcal{V}_{\mathrm{test}}|} \sum_{v \in \mathcal{V}_{\mathrm{test}}} \mathbf{1}[f(\mathbf{A}, \mathbf{X}, v) = y_v]. \tag{11}$$

C.2.3 Per-Probe Detection Scores and Structural Footprint

Statistical significance (SS, ours). SS is the $\log_{10}$ of the per-key Fisher-combined p-value p_{Fisher} (Algorithm 2, §3.3):

$$\mathrm{SS} := \log_{10} p_{\mathrm{Fisher}}, \tag{12}$$

computed under the benign null H_0. Smaller (more negative) values are stronger evidence against H_0. Unlike trigger baselines' binary fire/silent decisions, SS is a continuous and formally admissible test statistic (Sablayrolles et al., 2020) that quantifies the strength of evidence against H_0.

The per-key fires when $\mathrm{SS} < \kappa$, with the Bonferroni-corrected threshold (Dunn, 1961)

$$\kappa = \log_{10}(\tau / B), \tag{13}$$

so that the probability of any false fire across all B keys on a benign suspect stays at most τ. Without correction, attribution collapses: each key fires with probability τ, and there are B keys. The expected false fires per benign suspect is therefore approximately $B \cdot \tau$ (e.g., $1000 \times 0.05 = 50$ at $\tau = 0.05$).

Verification Success Rate (VSR, trigger baselines). VSR is the canonical detection score for trigger baselines (Backdoor, ImpWM, BkdOwn): the fraction of the trigger node set V_w predicted as the designated target label y^*:

$$\mathrm{VSR}(f) = \frac{1}{|V_w|} \sum_{v \in V_w} \mathbf{1}[f(\mathbf{A}, \mathbf{X}_{\mathrm{wm}}, v) = y^*]. \tag{14}$$

V_w is method-specific: trigger-carrying nodes for Backdoor; injected trigger nodes for ImpWM; centre nodes of trigger cliques for BkdOwn. A trigger fires when $\mathrm{VSR} > \kappa$. The chance-level threshold is $\kappa = 1/C$ for C classes, the probability of correctly predicting any specific label by uniform random guess.

Table 4: Dataset statistics.

Dataset	Homophily Ratio	#Nodes	#Edges	#Features	#Classes	Type
Ogbn-Arxiv (Hu et al., 2020)	0.65	169,343	1,157,799	128	40	Real world
Ogbn-Products (Hu et al., 2020)	0.81	2,449,029	61,859,012	100	47	Real world
Reddit (Hamilton et al., 2017)	0.76	232,965	57,307,946	602	41	Real world
Amazon-Ratings (Pei et al., 2020)	0.38	24,492	93,050	300	5	Real world
Roman-Empire (Pei et al., 2020)	0.05	22,662	32,927	300	18	Real world

Area Under the ROC Curve (AUC). AUC is the probability that a random watermarked suspect scores higher than a random benign suspect. $\mathrm{AUC}=1$ means perfect separation; $\mathrm{AUC}=0.5$ means the score is no better than random guessing. We train $n_+=n_-=10$ watermarked and benign suspects per method, assign each a score s (larger = stronger watermark evidence), and compute:

$$\mathrm{AUC}=\frac{1}{n_+n_-}\sum_{i:\,z_i=1}\sum_{j:\,z_j=0}\mathbf{1}[s_i>s_j]\in[0.5,1]. \tag{15}$$

AUC reflects per-method separability only: once a method's scores fully separate, AUC saturates at 1.0 and stops discriminating methods. We use the cross-method comparator TPR/FPR (below) for the saturated regime.

True/False Positive Rate at κ (TPR/FPR @ κ). TPR/FPR @ κ anchors all methods to a common firing threshold, providing the cross-method comparator that AUC cannot. On the same n_+ watermarked and n_- benign suspects, we count fires at κ:

$$\mathrm{TPR@}\kappa=\frac{k_+@\kappa}{n_+},\qquad \mathrm{FPR@}\kappa=\frac{k_-@\kappa}{n_-}, \tag{16}$$

where $k_+@\kappa$ and $k_-@\kappa$ are the numbers of watermarked and benign suspects that fire at κ, respectively. TPR/FPR @ κ resolves AUC's saturation: when methods all reach $\mathrm{AUC}=1$, TPR/FPR @ κ can still differ.

Structural footprint. As a structural counterpart to feature-space stealthiness (§C.2.1), we additionally report the triple $(|V_w|,\Delta N,\Delta E)$: the number of modified original nodes, injected new nodes, and injected edges in the released graph relative to the original. The triple captures how much the watermark perturbs graph structure, complementing the feature-space perturbation magnitude $(\mathcal{S},\bar{\mathcal{S}})$ and distributional distinctness $(\mathcal{S}_{\mathrm{knn}})$; smaller values indicate a stealthier structural footprint.

C.3 Experimental Settings

Datasets. Description and statistics for the five evaluation graphs are in Table 4.

Hardware and software. All experiments run on a single NVIDIA Quadro RTX 8000 GPU (46 GB memory), using PyTorch 2.2 and PyTorch Geometric 2.7.

Model training. The surrogate f_s and suspect models f_θ are optimised with Adam using learning rate 0.01, weight decay 5×10^{-4}, and dropout 0.5 for up to 1000 epochs, retaining the checkpoint with the best validation accuracy.

Ours. Our method's injection and detection procedures follow. (i) **Watermark injection**: we sample C per-class private keys uniformly from the unit sphere, kept secret by the owner. We select $|V_w|$ WM nodes class-balanced at random, with $|V_w|$ tuned per dataset to satisfy the fidelity budget. On each WM node v, we optimise a feature perturbation $\Delta\mathbf{x}_v$ to drive the per-class alignment $\cos(\Delta\mathbf{H}_v,u_{y_v})$ to the target cosine $\theta=0.96$, using momentum-SGD at learning rate 0.5 and momentum 0.99 for up to 3000 steps followed by L-BFGS fine-tuning for up to 150 steps. We use surrogate depth $K=1$ with no activation, perturbation bound $\delta=0.5$, and loss weights $\lambda_1=\lambda_2=0.1$. Per-dataset values are tuned for θ and effective-rank tolerance tol controlling k_{eff} from Corollary 1, together with the surrogate embedding dimension d_s, set equal to the node feature dimension so that the private keys live in the same space as the features. (ii) **Detection**: white-box detection computes the cosine score between the suspect's classifier weight and per-class private keys, while black-box detection runs a two-stage protocol. Stage 1 picks the candidate dataset $\hat{i}$ with the largest mean loss-gap on per-buyer V_w, committing to the watermarked dataset $\mathbf{D}^{(\hat{i})}_{\mathrm{wm}}$ if the peak gap exceeds threshold $\tau_d=0.5$ and to the benign dataset $\mathbf{D}$ otherwise. Stage 2 distills the suspect into a fixed 2-layer GCN student via two

queries on $\mathbf{x}_{\text{orig}}$ and the selected $\mathbf{x}_{\text{wm}}^{(\hat{i})}$ by matching the per-V_w-node logit shift via mean squared error (MSE) plus cross-entropy on $\mathbf{x}_{\text{orig}}$ (Eq. 53), trained over 1500 Adam epochs with cosine learning-rate schedule from 10^{-3}. The same cosine score is then applied on the student's classifier weight. Both aggregate per-class p-values via Fisher's method into $\log_{10} p_{\text{Fisher}}$. (iii) **Attribution**: at marketplace scale we generate B independent private-key sets and attribute the buyer with the smallest $\log_{10} p_{\text{Fisher}}$ below the Bonferroni threshold $\kappa = \log_{10}(0.05/B) \approx -4.30$ for $B = 1000$. For Amazon-Ratings ($C = 5$), a consistency penalty $\lambda = 10$ from Empirical Calibration 1 (Eq. 50), is added to suppress single-class outliers.

All baselines share with Ours the dataset splits, fidelity budget $|\Delta\text{Acc}| \leq 1\%$, marketplace scale $B = 1000$, and the same 50 benign reference suspects. Each method then trains its own 50 watermarked suspects, so cross-method differences reflect each method's watermark mechanism, per-buyer adaptation, and detection statistic alone. The parameter settings for each baseline are listed below.

(i) **RandPerturb**, an ablation of Ours: we sample per-class private keys as in Ours. We select $|V_w|$ watermark (WM) nodes class-balanced at random, with $|V_w|$ tuned per dataset to satisfy the fidelity budget $|\Delta\text{Acc}| \leq 1\%$. On each WM node v, we add random feature noise $\Delta\mathbf{x}_v$ with coordinates independently sampled from a uniform distribution on $[-\epsilon, \epsilon]$, producing $\mathbf{x}_v^{\text{wm}} = \mathbf{x}_v + \Delta\mathbf{x}_v$ with labels unchanged. The perturbation bound $\epsilon = \delta = 0.5$ matches Ours, isolating the absence of private-key guidance as the only difference. **Detection**: detection uses Ours' cosine-Fisher procedure. **Attribution**: at marketplace scale we generate a pool of B independent random perturbations, each paired with a buyer-specific private key, and attribute the buyer with the smallest $\log_{10} p_{\text{Fisher}}$ below the Bonferroni-corrected threshold $\kappa = \log_{10}(0.05/B) \approx -4.30$ for $B = 1000$.

(ii) **Backdoor** (Xu et al., 2023): we build a single trigger of length $\ell = 20$, with all ℓ dimensions set to the same value drawn uniformly from $[0, 1]$. We select $|V_w|$ WM nodes class-balanced at random, with $|V_w|$ tuned per dataset to satisfy the fidelity budget. On each WM node, we overwrite the ℓ trigger dimensions with the trigger value and relabel the node to target class 0. **Detection**: detection queries the suspect with each buyer's trigger applied to test nodes, computes the verification success rate (VSR) as the fraction predicted as target class 0, and flags the watermark as detected when VSR exceeds the per-class baseline $1/C$. **Attribution**: at marketplace scale we generate a pool of B distinct triggers by sampling a unique trigger value per buyer, and attribute to the buyer with the highest VSR among the B trigger queries.

(iii) **BkdOwn** (Li et al., 2023): we construct a 4-node trigger clique whose nodes share a feature vector obtained by perturbing the training-set feature mean with a scale-0.5 shift along a random unit direction kept as the owner's secret. We select $|V_w|$ WM nodes by class-balanced random sampling, treating each WM node's k-hop subgraph as an individual graph as in BkdOwn's graph-classification setup, with $|V_w|$ tuned per dataset to satisfy the fidelity budget. On each subgraph, we inject the trigger clique by connecting all four clique nodes to the WM node, and relabel the WM node to target class 0. The suspect is trained as a graph classifier over these labelled subgraphs. **Detection**: detection queries the suspect on each WM node's k-hop subgraph with the buyer's clique attached, computes the verification success rate (VSR) as the fraction predicted as target class 0, and flags the watermark as detected when VSR exceeds the per-class baseline $1/C$. **Attribution**: at marketplace scale we generate a pool of B distinct trigger cliques by sampling a different random unit direction per buyer, and attribute to the buyer with the highest VSR among the B trigger queries.

(iv) **ImpWM** (Zhang et al., 2024a): the trigger is a structured subgraph of $|V_w|$ new nodes, with $|V_w|$ tuned per dataset to satisfy the fidelity budget. We select C anchor nodes, one per class, and relabel them to target class 0. Each new node connects to one anchor by a single edge and is labelled as target class 0 too. New nodes interconnect via 3 rounds of derangement. Each new node's features are produced by a trainable GCN generator that is optimised so that trigger features mimic anchor features for imperceptibility, owner identity is encoded in the last 5 dimensions via binary cross-entropy against a per-buyer owner-id vector, and a pre-trained 2-layer GCN reference model predicts target class 0 on new nodes. For data-IP adaptation, we train only the GCN generator as outer-level optimisation and keep the reference model fixed in place of the original bi-level inner-loop suspect retraining. **Detection**: detection queries the suspect on the $|V_w|$ new nodes and the C anchors, computes the verification success rate (VSR) as the fraction predicted as target class 0, and flags the watermark as detected when VSR exceeds the per-class baseline $1/C$. **Attribution**: at marketplace scale we generate a pool of B distinct trigger subgraphs by sampling a per-buyer owner-id vector

and random derangement seed, and attribute to the buyer with the highest VSR among the B trigger queries.

(v) **Data Taggants** (Bouaziz et al., 2025): we randomly sample the original paper's $K = 10$ key vectors of the node feature dimension from a zero-mean Gaussian and normalise them to the unit sphere as in Ours' private-key sampling, each paired with a random target label, kept secret. We split the $|V_w|$ WM nodes equally across the K keys, drawing each key's $|V_w|/K$ nodes at random from all graph nodes whose true label matches that key's label, as the signer cannot control the suspect's train/test split, with $|V_w|$ tuned per dataset to satisfy the fidelity budget. On each WM node, only the features are perturbed within max-norm bound $\delta = 0.5$, matching Ours' perturbation bound. The shift $\Delta\mathbf{x}_v$ is optimised to the target average cosine $\theta = 0.96$ across the K keys between per-key parameter-gradients on a 1-layer linear surrogate reference, the same surrogate model used by Ours. For each key, the WM-side gradient is cross-entropy on its WM nodes against their true labels, and the key-side gradient is cross-entropy on the key against its key label. Optimisation uses momentum SGD with coefficient 0.99 on the unit-normalised gradient at learning rate 0.5 for up to 3000 steps, with max-norm projection after each step. **Detection**: detection queries the suspect on each buyer's K keys, counts X keys where the assigned key label appears in the suspect's top-k predictions, computes the binomial p-value under null $X \sim \text{Bin}(K, k/C)$, and flags the watermark as detected when $\log_{10} p < \kappa = \log_{10}(0.05) \approx -1.30$. We set the top-$k$ following the original paper, which targets a null of about 0.01 and reaches it with $k = 10$ at $C = 1000$. Our graph benchmarks all have $C \leq 47$, so $\text{round}(0.01\, C)$ floors to $k = 1$ on every dataset, the smallest attainable null $1/C$. **Attribution**: at marketplace scale each of B buyers has their own secret K-key set, and we apply the Detection binomial test separately to each buyer's keys to obtain a per-buyer $\log_{10} p_b$. We attribute to the buyer with the smallest $\log_{10} p_b$, requiring it to fall below the Bonferroni-corrected threshold $\kappa = \log_{10}(0.05/B) \approx -4.30$ for $B = 1000$.

D Theory, Proofs, and Empirical Validation

- §D.1 characterizes the null distribution of the cosine similarity between two uniformly random unit vectors, providing the basis for per-class significance testing.
- §D.2 establishes the direction of the alignment matrix that maps surrogate embeddings to suspect embeddings, distinguishing forward from backward inversion.
- §D.3 formally defines the shared subspace and its effective rank, and establishes that classifier weights lie in this subspace by Property 1 and that the subspace is architecture-invariant by Property 2.
- §D.4 constructs the whitening operator that projects and calibrates both the classifier weights and the private key into the shared subspace before computing the per-class alignment score.
- §D.5 applies this alignment score as the input to Proposition 1's null distribution, deriving the Fisher-combined detection statistic and Corollary 1's high-probability sufficient condition for detection.
- §D.6 proves Proposition 2, establishing a bound on the effective rank that delimits the operational range under which Corollary 1's condition remains valid.
- §D.7 extends the single-key detection analysis to the multi-buyer setting, deriving Corollary 2's approximate joint detection-and-attribution success rate and introducing the consistency-penalised statistic of Empirical Calibration 1 that tightens this approximation at low effective rank.

D.1 Cosine Null Distribution (Proposition 1, restated)

The closed-form null distribution of cosine similarity between two random unit vectors in $\mathbb{R}^d$ is a classical result from directional statistics (Mardia & Jupp, 2009) and has been used in fingerprinting and watermarking contexts by (Iscen et al., 2018; Sablayrolles et al., 2020). We restate it here for completeness as it underlies our per-class p-value (§3.3) and the detection scaling law (Theorem 1).

Proposition 1 (Null distribution of cosine alignment (Mardia & Jupp, 2009; Iscen et al., 2018; Sablayrolles et al., 2020))**.** *Let $\mathbf{v} \in \mathbb{R}^d$ be a fixed unit vector and $\mathbf{u} \in \mathbb{R}^d$ drawn uniformly at random from the unit sphere (i.e., independent of $\mathbf{v}$). Then the cosine similarity $\cos(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v}$*

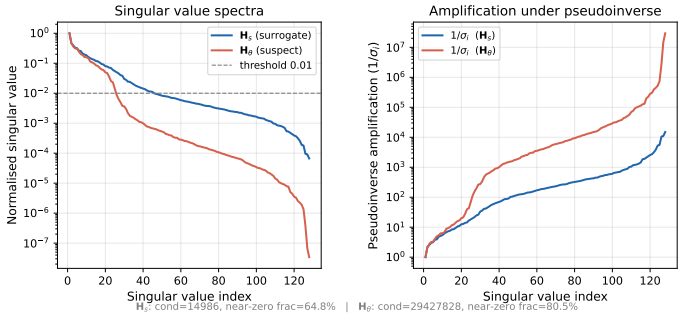


Figure 4: Singular value spectra (left) and pseudoinverse amplification $1/\sigma_i$ (right) for $\mathbf{H}_s$ and $\mathbf{H}_\theta$ on 53,877 hold-out nodes. $\mathbf{H}_\theta$ collapses far more severely, making its pseudoinverse numerically unreliable.

has expectation 0*, variance* $1/d$*, and its tail probability for an observed value* $t \geq 0$ *satisfies*

$$\mathbb{P}\big(\cos(\mathbf{u}, \mathbf{v}) \geq t\big) = \frac{1}{2} I_{1-t^2}(a,\, b)\,, \tag{17}$$

where $I_x(a, b)$ *is the regularised incomplete beta function,* $a = (d-1)/2$ *and* $b = 1/2$ *are shape parameters determined by the vector dimension* d*, and* t *denotes the observed cosine similarity between model weights and carrier direction.*

D.2 Singular Value Spectra of $\mathbf{H}_s$ and $\mathbf{H}_\theta$

This section aims to identify the alignment direction under which the downstream cosine detection $\cos(W_c, u_c)$ actually fires on the watermark signal. Section 3.3 introduces this alignment step between the surrogate embedding $\mathbf{H}_s$ and the suspect embedding $\mathbf{H}_\theta$. Two directions of alignment are possible:

- **Forward** ($\mathbf{X}_{\text{align}} = \mathbf{H}_s^\dagger \mathbf{H}_\theta$): inverts $\mathbf{H}_s$ via least-squares regression.
- **Backward** ($\mathbf{X}_{\text{align}} = \mathbf{H}_\theta^\dagger \mathbf{H}_s$): inverts $\mathbf{H}_\theta$.

Choosing between them reduces to which of $\mathbf{H}_s$, $\mathbf{H}_\theta$ is safer to invert, and this in turn is determined by their singular value spectra. The pseudoinverse $\mathbf{H}^\dagger$ re-weights each singular direction by $1/\sigma_i$, so a steep-decay spectrum blows up $1/\sigma_i$ on weak directions, amplifying noise over signal and leaving $\mathbf{X}_{\text{align}}$ dominated by noise. We therefore compare the singular value spectra of the surrogate embedding matrix $\mathbf{H}_s$ and the suspect model embedding matrix $\mathbf{H}_\theta$, both computed on the $N_{\text{fit}} = 53{,}877$ hold-out validation and test nodes of ogbn-arxiv, to identify which side is invertible-safe.

Figure 4 (left) plots normalised singular values σ_i/σ_1 (log scale): $\mathbf{H}_\theta$ (suspect, 2-layer GCN) collapses much faster than $\mathbf{H}_s$ (surrogate, 1-layer linear GNN): 80.5% vs 64.8% of values fall below $10^{-2}\sigma_1$, and the condition number of $\mathbf{H}_\theta$ ($\approx 2.9 \times 10^7$) is $\mathbf{2000}\times$ that of $\mathbf{H}_s$ ($\approx 1.5 \times 10^4$). Figure 4 (right) shows the resulting pseudoinverse amplification $1/\sigma_i$ blows up by orders of magnitude on $\mathbf{H}_\theta$, making $\mathbf{H}_\theta^\dagger$ numerically unreliable while $\mathbf{H}_s^\dagger$ stays well-conditioned.

Detection failure under the backward direction. To verify the consequence, we compare two alignment directions on ogbn-arxiv across all tolerance levels (Table 5): **Forward** ($\mathbf{H}_s^\dagger \mathbf{H}_\theta$, ours) inverts the well-conditioned $\mathbf{H}_s$, while **Backward** ($\mathbf{H}_\theta^\dagger \mathbf{H}_s$, alternative) inverts the ill-conditioned $\mathbf{H}_\theta$; we additionally run Forward on a clean (unwatermarked) model to verify no false accusation. These three outcomes together decide the alignment direction:

- **Backward on the watermarked model** yields $\log_{10} p \approx 0$ at every tolerance: inverting the ill-conditioned $\mathbf{H}_\theta$ amplifies noise through $1/\sigma_i$, so $\mathbf{X}_{\text{align}}$ loses signal and the cosine test cannot fire on the watermark carrier. This shows Backward is *not viable*.
- **Forward on the clean (unwatermarked) model** yields $\log_{10} p \approx 0$: with no injected signal, the alignment scores are uniform under $\mathcal{H}_0$, confirming Forward does not fabricate spurious alignment on innocent suspects. This shows Forward is *safe*.

Table 5: Alignment directions and models on ogbn-arxiv. k_{eff}: effective rank after spherical whitening (§3.3); "Mean cos": average cosine across marked classes (expected 0 under H_0).

	Forward, Watermarked			**Backward, Watermarked**			**Forward, Benign**		
Tolerance	k_{eff}	Mean cos	$\log_{10}(p)$	k_{eff}	Mean cos	$\log_{10}(p)$	k_{eff}	Mean cos	$\log_{10}(p)$
10^{-1}	14	0.53	-36.9	3	0.11	$\mathbf{-1.6}$	13	0.00	-0.5
10^{-2}	27	0.42	$\mathbf{-43.37}$	8	-4.40	-0.2	25	0.00	-0.4
10^{-3}	46	0.29	-37.6	27	-4.20	0.0	41	-1.40	-0.1
10^{-4}	80	0.11	-10.9	56	0.00	-0.2	67	0.03	-0.8

- **Forward on the watermarked model** achieves $\log_{10} p = -43.4$ at 10^{-2}: inverting the well-conditioned $\mathbf{H}_s$ preserves the signal in $\mathbf{X}_{\text{align}}$, so the cosine test recovers the injected watermark carrier and fires strongly. This shows Forward is *effective*.

Section 3.3's alignment matrix $\mathbf{X}_{\text{align}} = \mathbf{H}_s^{\dagger}\mathbf{H}_\theta$ therefore adopts the **Forward** direction: the empirical evidence here validates that Forward is necessary (Backward fails), safe (no false accusation on clean models), and effective (strong detection on watermarked models).

D.3 Formal Definition of the Shared Subspace S

Definition. For a fixed surrogate model with embedding function ϕ_s and any suspect t with embedding function ϕ_t, let $\mathbf{H}_s := \phi_s(V_{\text{fit}}) \in \mathbb{R}^{N_{\text{fit}} \times d}$ and $\mathbf{H}_\theta^{(t)} := \phi_t(V_{\text{fit}}) \in \mathbb{R}^{N_{\text{fit}} \times d_{\mathbf{H}}}$ be the embeddings on the alignment node set V_{fit}, consisting of validation and test nodes excluding watermarked carriers. The pair-specific alignment matrix is

$$\mathbf{X}_{\text{align}}^{(t)} := \arg\min_{\mathbf{X} \in \mathbb{R}^{d \times d_{\mathbf{H}}}} \left\| \mathbf{H}_s \mathbf{X} - \mathbf{H}_\theta^{(t)} \right\|_F^2 = \mathbf{H}_s^{\dagger}\, \mathbf{H}_\theta^{(t)}. \tag{18}$$

Let $\mathbf{X}_{\text{align}}^{(t)} = \mathbf{U}_t \mathbf{\Sigma}_t \mathbf{V}_t^{\top}$ be its singular value decomposition (SVD) with $\mathbf{V}_t = [\mathbf{v}_1^{(t)}, \ldots, \mathbf{v}_{d_{\mathbf{H}}}^{(t)}]$ and singular values $\sigma_1^{(t)} \geq \cdots \geq \sigma_{d_{\mathbf{H}}}^{(t)} \geq 0$. For an SVD relative tolerance $\text{tol} \in (0, 1)$, the *effective rank* is $k_{\text{eff}}(t) := |\{j : \sigma_j^{(t)} > \text{tol}\, \sigma_1^{(t)}\}|$, and the *shared subspace* in the suspect's frame is

$$S^{(t)} := \text{span}\big(\{\mathbf{v}_j^{(t)} : j \leq k_{\text{eff}}(t)\}\big) \subset \mathbb{R}^{d_{\mathbf{H}}}. \tag{19}$$

$S^{(t)}$ is pair-specific in suspect frame, with its basis $\mathbf{V}_k^{(t)}$ varies across suspects. Property 2 shows that the matched surrogate-frame structure $\mathbf{U}_k^{(t)} \subset \mathbb{R}^d$ is approximately invariant across suspects, and through $\mathbf{X}_{\text{align}}^{(t)}$ this invariance carries to each suspect's $S^{(t)}$. We therefore drop the superscript and refer to a single conceptual S when context permits.

Motivation. Our cosine-based detection test computes $\cos(W_c, u_c)$ inside the subspace S. For this to be meaningful, two questions must be answered: (a) does the trained classifier weight W_c actually live in S (otherwise the cosine measures noise, not signal)? and (b) is S the same geometric object across different suspect architectures (otherwise cross-architecture attribution breaks)? Properties 1 and 2 answer these two questions:

- **Property 1** justifies restricting detection to the subspace S at all (the signal lives there, not in the full embedding space). This underlies the *validity* of the test.
- **Property 2** justifies that this S is approximately the same geometric object regardless of the suspect's architecture. This underlies the *architecture-agnostic* claim, arguably the paper's headline property.

Together they bridge the informal geometric intuition of Figure 2 (§3.1) to the precise machinery used later: §D.4 builds the whitening matrix $\mathbf{M}$ from this S, and §D.5's Theorem 1 / Corollary 1 (the detection guarantee) is stated directly in terms of k_{eff}, the dimension of this S.

Property 1: Classifier weights lie in $S^{(t)}$. For a classifier f_θ trained on the labeled dataset under any prediction-based loss L whose dependence on $\mathbf{W}_c$ enters only through per-sample logits $\mathbf{W}_c \cdot \mathbf{h}_n$,

the per-class weight $\mathbf{W}_c$ is approximately contained in $S^{(t)}$. Concretely, the chain rule gives the gradient on $\mathbf{W}_c$ as a linear combination of node embeddings:

$$\partial L/\partial \mathbf{W}_c \;=\; \sum_n \alpha_n \, \mathbf{h}_n,$$

where the sum runs over training nodes n and $\alpha_n \in \mathbb{R}$ is a per-node scalar coefficient determined by L. For cross-entropy, $\alpha_n = p_n - \mathbf{1}[y_n = c]$, where $p_n := \mathrm{softmax}(\mathbf{W}\mathbf{h}_n)_c$ is the model's predicted probability for class c and y_n is the true label. Squared error, margin/hinge, focal, and label-smoothed losses yield different α_n with the same linear-combination form. Since the task confines $\{\mathbf{h}_n\}$ to a low-dimensional discriminative span and $\mathbf{X}^{(t)}_{\text{align}}$'s top-$k_{\text{eff}}$ right singular subspace captures exactly this span in the suspect's frame, each $\mathbf{h}_n$ lies in $S^{(t)}$. By linearity, the gradient (a linear combination of $\{\mathbf{h}_n\}$) also lies in $S^{(t)}$, so every gradient step updates $\mathbf{W}_c$ along directions inside $S^{(t)}$ and keeps $\mathbf{W}_c$ within $S^{(t)}$ up to a small residual from random initialisation. Equivalently, the energy fraction $\beta^{(c)}_W := \|P_{S^{(t)}}(\mathbf{W}_c)\|^2/\|\mathbf{W}_c\|^2$ concentrates a majority of $\mathbf{W}_c$'s squared norm inside $S^{(t)}$; empirically β_W has mean 0.83 (observed range $[0.55, 1.00]$) across 30 mark suspects on arxiv spanning GCN, GAT, and GraphSAGE architectures.

Only $P_{S^{(t)}}(u_c)$ drives alignment and the orthogonal complement is noise. Decomposing $u_c \in \mathbb{R}^d$ into its in-$S^{(t)}$ and orthogonal components, $u_c = P_{S^{(t)}}(u_c) + P_{S^{(t)\perp}}(u_c)$, the alignment splits additively as

$$\mathbf{W}_c \cdot u_c \;=\; \underbrace{\mathbf{W}_c \cdot P_{S^{(t)}}(u_c)}_{\text{signal: aligned by training}} \;+\; \underbrace{\mathbf{W}_c \cdot P_{S^{(t)\perp}}(u_c)}_{\text{noise: bounded}}. \tag{20}$$

By Property 1, $\mathbf{W}_c$ resides almost entirely in $S^{(t)}$ with energy fraction β_W, so its component outside $S^{(t)}$ has norm $\sqrt{1-\beta_W}$. Applying Cauchy–Schwarz to the second term,

$$|\mathbf{W}_c \cdot P_{S^{(t)\perp}}(u_c)| \;\le\; \sqrt{1-\beta_W} \cdot \|P_{S^{(t)\perp}}(u_c)\|,$$

which is structurally noise rather than signal because (i) the training objective induces no preferred direction in $S^{(t)\perp}$ for $\mathbf{W}_c$, so $P_{S^{(t)\perp}}(\mathbf{W}_c)$ and $P_{S^{(t)\perp}}(u_c)$ behave as independent random vectors in a $(d - k_{\text{eff}})$-dimensional space and their cosine concentrates at zero with standard deviation $1/\sqrt{d-k_{\text{eff}}}$; and (ii) on dataset ogbn-arxiv with $\beta_W \ge 0.6$, $d = 128$, $k_{\text{eff}} = 17$, the typical orthogonal-term magnitude $\sqrt{1-\beta_W} \cdot 1/\sqrt{d-k_{\text{eff}}} \le 0.06$ is an order of magnitude below the signal-term magnitude $\|P_{S^{(t)}}(u_c)\| \approx \sqrt{k_{\text{eff}}/d} \approx 0.36$ obtained by Property 1 plus the standard spherical-marginal result (Vershynin, 2026). Watermark detection therefore activates alignment only from $P_{S^{(t)}}(u_c)$; the orthogonal component contributes uncorrelated noise that the Fisher combination across C classes averages out. Empirically on ogbn-arxiv with a 1-layer linear surrogate and an $N=2$ GCN suspect, the alignment test restricted to the in-$S^{(t)}$ component $P_{S^{(t)}}(u_c)$ yields $\log_{10} p \approx -43$ on watermarked suspects against $\log_{10} p \approx 0$ on benign FP probes; without this restriction the orthogonal component $P_{S^{(t)\perp}}(u_c)$ inflates the null and triggers benign FP over-firing, confirming the signal-noise split in Eq. 20.

Property 2: Cross-architecture invariance. The shared subspace S is approximately the same across suspect architectures. Although $S^{(t)}$ is formally defined via $\mathbf{V}^{(t)}_k$ in the suspect frame (Eq. 19), its identity is best diagnosed via $\mathbf{U}^{(t)}_k$ in the common surrogate frame $\mathbb{R}^d$. The SVD of the alignment matrix makes this concrete on its two sides. The top-k_{eff} *left* singular vectors $\mathbf{U}^{(t)}_k \subset \mathbb{R}^d$ (in the surrogate frame) are nearly identical across all suspects, because they are pinned by the discriminative geometry of the labeled dataset, which every suspect shares regardless of its own architecture or training trajectory. The right singular vectors $\mathbf{V}^{(t)}_k \subset \mathbb{R}^{d_{\mathbf{H}}}$ (in each suspect's own embedding frame) do change with the suspect, but each $\mathbf{V}^{(t)}_k$ is just the same $\mathbf{U}^{(t)}_k$ pushed through $\mathbf{X}^{(t)}_{\text{align}}$ into the suspect's frame, so all suspects encode one underlying S in different coordinates. Since $\mathbf{\Sigma}$ only rescales and $\mathbf{X}^{(t)}_{\text{align}}$ transports without introducing new directions, $\mathbf{U}_k$ alone specifies the suspect-independent identity of S. This is why the owner can approximately fix S at design time from the surrogate alone and then use it for any future suspect. Define the index-j principal cosine on the surrogate side as

$$\rho_j \;:=\; \mathbb{E}_{(t_1,t_2)\sim\mathcal{P}}\big[\cos\big(\mathbf{u}^{(t_1)}_j, \mathbf{u}^{(t_2)}_j\big)\big],$$

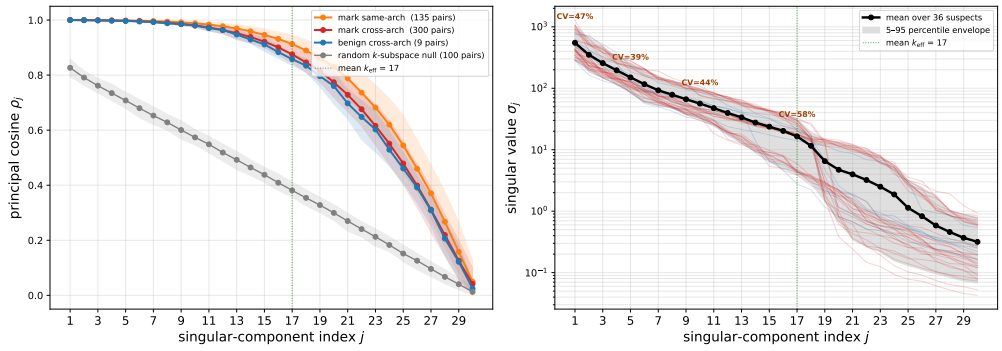


Figure 5: Empirical verification of Property 2 on arxiv. **Left**: index-j principal cosines ρ_j between top left singular vectors $\mathbf{U}_j^{(t)}$ of $\mathbf{X}_{\text{align}}^{(t)}$ across suspect pairs. **Right**: per-suspect singular value spectra $\sigma_j^{(t)}$ of $\mathbf{X}_{\text{align}}^{(t)}$.

where $\mathcal{P}$ is an empirical pair distribution.

Empirically, we instantiate $\mathcal{P}$ on ogbn-arxiv with 30 trained mark suspects: 10 2-layer GCN, 10 3-layer GAT, and 10 3-layer GraphSAGE, each architecture spanning seeds 42–51. All suspects are trained on the same watermarked dataset, and the surrogate frame is fixed to a seed-42 1-layer linear model. The aggregated results (Figure 5, left) show: (i) the top singular direction is essentially identical across all architectures ($\rho_1 = 1.000$); (ii) the top 10 directions are shared to > 0.98 regardless of architecture; (iii) at the effective-rank boundary $j = k_{\text{eff}} \approx 17$, cross-architecture overlap remains $\rho_{17} \geq 0.87$ across 300 cross-architecture pairs, with the same-architecture upper bound at $\rho_{17} = 0.913$; (iv) for all $j \leq k_{\text{eff}}$, the empirical cosines stay orders of magnitude above the random-subspace baseline (grey curve). This evidence empirically justifies treating $\mathbf{U}_k^{(t)}$ as approximately a single conceptual subspace S in the surrogate frame, with each suspect's $\mathbf{V}_k^{(t)}$ serving as its frame-specific image under $\mathbf{X}_{\text{align}}^{(t)}$. In contrast, Figure 5 (right) shows that the per-suspect singular magnitudes $\sigma_j^{(t)}$ are not invariant: their coefficient of variation across the 36 suspects is 40–60% at every index $j \leq k_{\text{eff}}$. Consequently, the $\mathbf{\Sigma}_k^{-1}$ factor in the detection-time whitening matrix $\mathbf{M} = \mathbf{V}_k \mathbf{\Sigma}_k^{-1} \mathbf{V}_k^\top$ rescales each in-S direction by $1/\sigma_j^{(t)}$: the same projection of u_c onto direction j therefore contributes a suspect-specific weight to the cosine alignment with $\mathbf{W}_c$. Hence $\mathbf{M}$ must be computed per suspect from $\mathbf{X}_{\text{align}}^{(t)}$ and cannot be reused from the fixed "universal S".

D.4 SVD-based Spherical Whitening Derivation

This section gives the full SVD-based derivation of the whitening matrix $\mathbf{M}$ used in §3.3. We expand $u_c^m = u_c \mathbf{X}_{\text{align}}$ along the SVD of $\mathbf{X}_{\text{align}}$, show how $\mathbf{M}$ cancels singular-value amplification on the top-k_{eff} subspace and projects out the noise-dominated complement, and explain the spectral rationale for low-rank truncation.

Why the whitening matrix is needed. Transporting the secret key u_c from the surrogate frame into the suspect frame via $u_c^m := u_c \mathbf{X}_{\text{align}}$ is not a clean rotation; it is a multiplication by a matrix with an arbitrary singular spectrum. Under the SVD of $\mathbf{X}_{\text{align}}$, each direction is scaled by its own singular value σ_j. Since u_c starts out uniformly distributed on the unit sphere, this per-direction rescaling distorts that uniformity: some directions are inflated, others suppressed. The consequence in §D.3 is that this anisotropy can make a *benign* $\mathbf{W}_c$ look spuriously aligned with u_c^m purely as an artifact of the singular-value spectrum, not because of any real watermark signal, inflating false positives if the raw transport were used. The whitening matrix $\mathbf{M}$ constructed below undoes this per-direction distortion on the top-k_{eff} subspace and discards the noise-dominated complement, restoring an approximately isotropic transported key.

SVD expansion of u_c^m. Decompose $\mathbf{X}_{\text{align}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_\mathbf{H}}]$ and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_{d_\mathbf{H}}]$ are orthonormal left/right singular vectors and $\mathbf{\Sigma} = \text{diag}(\sigma_1 \geq \cdots \geq \sigma_{d_\mathbf{H}})$. Each direction is amplified by σ_j: the projection of u_c^m along the j-th right singular vector $\mathbf{v}_j$ equals

Table 6: Empirical mean pairwise Pearson correlation of $\{s_c\}$ under $\mathcal{H}_0$, computed across 50 suspects $\times$ 1000 carrier keys per dataset (50,000 realisations); $|\rho_{ij}|$ averaged over all $C(C-1)/2$ off-diagonal pairs.

Dataset	C	mean $\lvert\rho_{ij}\rvert$
ogbn-arxiv	40	0.125
Amazon-Ratings	5	0.224
Reddit	41	0.134
Roman-Empire	18	0.154
ogbn-products	47	0.124

$(u_c \cdot \mathbf{u}_j)\,\sigma_j$. Expanding:

$$u_c^m \;=\; \sum_{j=1}^{d_{\mathbf{H}}} (u_c \cdot \mathbf{u}_j)\,\sigma_j\,\mathbf{v}_j^\top. \tag{21}$$

Highly unequal σ_j inflate certain directions and suppress others, breaking the uniform spherical distribution of u_c on the unit sphere; this anisotropy can cause a benign $\mathbf{W}_c$ to appear spuriously aligned with u_c^m even without any watermark signal, inflating false positives, as empirically illustrated for the in-S vs out-of-S split following Eq. 20.

Constructing the whitening matrix. We restore u_c^m by constructing $\mathbf{M} = \mathbf{V}_k \mathbf{\Sigma}_k^{-1} \mathbf{V}_k^\top$, where $\mathbf{V}_k \in \mathbb{R}^{d_{\mathbf{H}} \times k_{\text{eff}}}$ retains the right singular vectors associated with the top k_{eff} singular values (those with $\sigma_j > \text{tol}\,\sigma_1$). Applying $\mathbf{M}$:

$$u_c^m\,\mathbf{M} \;=\; \sum_{j=1}^{k_{\text{eff}}} (u_c \cdot \mathbf{u}_j)\,\underbrace{\sigma_j \cdot \tfrac{1}{\sigma_j}}_{=1}\,\mathbf{v}_j^\top \;+\; \underbrace{\sum_{j=k_{\text{eff}}+1}^{\min(d,\,d_{\mathbf{H}})} 0}_{\text{discarded}} \;=\; u_c\,\mathbf{U}_k \mathbf{V}_k^\top. \tag{22}$$

$\mathbf{M}$ does two things simultaneously: for each of the k_{eff} retained directions, it multiplies by $1/\sigma_j$, cancelling the σ_j amplification; for the remaining $\min(d, d_{\mathbf{H}}) - k_{\text{eff}}$ directions, $\mathbf{M}$ projects them to zero (since $\mathbf{V}_k$ spans only the first k_{eff} directions). The result is a clean cross-space map $\mathbf{U}_k \mathbf{V}_k^\top$.

D.5 Detection Guarantee and its Proof

Theorem 1 (Detection strength)**.** *Let C be the number of classes and $\{s_c\}_{c=1}^C$ the per-class alignment scores under $\mathcal{H}_0$. Assume (i) the $\{s_c\}$ are approximately independent and (ii) the effective rank $k_{\text{eff}} \geq 3$. Then*

$$-\ln p_{\text{Fisher}} \;=\; \frac{k_{\text{eff}}}{2} \sum_{c=1}^{C} s_c^2 \;+\; R_{\text{Fisher}}(k_{\text{eff}}, \{s_c\}), \tag{23}$$

where R_{Fisher} is the explicit residual defined in Eq. 32 (Appendix D.5), bundling per-class Taylor remainders, sub-leading corrections, and negatively-aligned class corrections.

Remark 1 (Correlation quantification for Theorem 1's assumption (i))**.** We measure the empirical Pearson correlation $\rho_{ij} := \text{Cov}(s_i, s_j)/(\sigma_{s_i}\,\sigma_{s_j})$ between per-class scores under $\mathcal{H}_0$, where Cov denotes covariance and σ_{s_i} the standard deviation of s_i, both computed empirically across 50 suspects $\times$ 1000 carrier keys per dataset (Table 6). Averaging $|\rho_{ij}|$ over all $C(C-1)/2$ off-diagonal pairs, the mean pairwise correlation magnitude ranges from 0.12 (products) to 0.22 (amazon), indicating a modest deviation from strict independence.

Non-zero $\bar{\rho}$ inflates Corollary 1's Bernstein margin(δ_B) by factor $\sqrt{1 + (C-1)\bar{\rho}}$, requiring a larger watermark-strength signal than under strict independence. The total variance grows from $C \cdot V$ (independence) to $C \cdot V \cdot (1 + (C-1)\bar{\rho})$ (average correlation $\bar{\rho}$); the margin scales as the square root of the variance (Maurer & Pontil, 2009). Derivation in Corollary 1's proof.

Remark 2 (Effective-rank k_{eff} tradeoff in Theorem 1)**.** Increasing k_{eff} amplifies the prefactor $k_{\text{eff}}/2$ but also admits noise directions that reduce $\sum_c s_c^2$; the resulting tradeoff is bounded by $3 \leq k_{\text{eff}} \leq C$ (upper bound from Assumption 1). The raw alignment $\cos(\mathbf{W}_c, u_c)$ in the suspect's embedding space is additionally capped at the geometric ceiling $\sqrt{k_{\text{eff}}/d}$ from the spherical-marginal result (Appendix D.3).

Table 7: Tolerance sweep on ogbn-arxiv (C=40). Leading term + residual term $= -\ln p_{\text{Fisher}}$ column-wise, with peak at k_{eff}=25.

tol	k_{eff}	vs C	leading term $(k_{\text{eff}}/2)\sum_c s_c^2$	residual term R_{Fisher}	$-\ln p_{\text{Fisher}}$
10^{-1}	13	$<$	74.9	$+4.3$	79.2
10^{-2}	**25**	$<$	**86.0**	$\mathbf{+12.0}$	**98.0**
$\approx 10^{-3}$	39	$<$	65.7	$+4.8$	70.5
5×10^{-4}	46	$>$	55.9	$+5.5$	61.4
10^{-4}	63	$\gg$	14.8	$+3.9$	18.7

Empirical validation. Theorem 1 splits $-\ln p_{\text{Fisher}}$ into a leading term $(k_{\text{eff}}/2)\sum_c s_c^2$ and a residual R_{Fisher}. Larger k_{eff} has two opposing effects: the prefactor $k_{\text{eff}}/2$ grows, but noise directions dilute $\sum_c s_c^2$. Table 7 on ogbn-arxiv (C=40) shows the leading term peaks at k_{eff}=25 (value 86.0, $-\ln p_{\text{Fisher}} = 98.0$), well below C=40. The leading term and R_{Fisher} sum to $-\ln p_{\text{Fisher}}$ at every row (Eq. 23), with $R_{\text{Fisher}} \in [+3.9, +12.0]$, far smaller than the leading term's range.

Proof of Theorem 1. We assume that models trained on the same data distribution learn partially aligned discriminative subspaces in the representation space, which is a standard assumption in representation similarity and model extraction literature (Raghu et al., 2017; Morcos et al., 2018; Kornblith et al., 2019). Under the alignment test, the per-class score $s_c = \tilde{u}_c \cdot \tilde{\mathbf{W}}_c \in [-1, 1]$ lives in a k_{eff}-dimensional space obtained by projecting onto the top singular directions of $\mathbf{X}_{\text{align}}$.

Step 1: Null distribution. Under $\mathcal{H}_0$, the whitening procedure (§3.3) maps both vectors into a k_{eff}-dimensional subspace. The carrier u_c is drawn uniformly on S^{D-1} and is independent of $\mathbf{W}_c$; by the rotational invariance of the uniform distribution, the whitened carrier $\tilde{u}_c$ is uniformly distributed on $S^{k_{\text{eff}}-1}$, while $\tilde{\mathbf{W}}_c$ is fixed. Proposition 1 therefore applies exactly: s_c follows $\text{CosineNull}(k_{\text{eff}})$ with mean 0 and variance $1/k_{\text{eff}}$.

Step 2: From score to p-value. Our test is *one-sided* (upper tail): only $s_c > 0$ constitutes evidence of watermarking, since the embedding procedure aligns perturbations with the carrier in the positive direction. Applying Proposition 1 with $d = k_{\text{eff}}$ (justified by Step 1), the exact one-sided p-value is

$$p_c \;=\; P(S \geq s_c \mid \mathcal{H}_0) \;=\; \tfrac{1}{2}\, I_{1-s_c^2}\Big(\tfrac{k_{\text{eff}}-1}{2},\, \tfrac{1}{2}\Big), \quad s_c > 0, \tag{24}$$

where $I_x(a, b)$ is the regularised incomplete beta function. For $s_c \leq 0$, the cosine null is symmetric about zero, so $P(S \leq s_c) = P(S \geq -s_c)$; hence $p_c = P(S \geq s_c) = 1 - P(S \geq |s_c|) \geq 1 - 1/2 = 1/2$.

Step 3: Fisher combination. Step 2 produces C per-class p-values $\{p_c\}_{c=1}^{C}$ from the exact cosine null (Eq. 24). Under $\mathcal{H}_0$, each p_c is uniformly distributed on $[0, 1]$, so $-2\ln p_c \sim \chi^2(2)$. In practice, the C class-wise scores share the same GNN backbone and classifier weight matrix, so they are not strictly independent; we treat them as approximately independent (Fisher, 1925), with the empirical deviation quantified in Remark 1. Pooling 50 benign suspects $\times$ 5 datasets $= 250$ total, the empirical false positive rate under the $\chi^2(2C)$-derived Bonferroni threshold $\ln(B/\tau)$ is $0/250 = 0\%$ (95% confidence upper bound $= 1 - 0.05^{1/250} \leq 1.2\%$), far below the nominal $\tau = 5\%$, empirically confirming the $\chi^2(2C)$ approximation error is negligible under the measured $\bar{\rho}$. Under this assumption, the combined statistic

$$T \;=\; -2\sum_{c=1}^{C} \ln p_c \;\dot{\sim}\; \chi^2(2C) \tag{25}$$

approximately follows $\chi^2(2C)$ under $\mathcal{H}_0$, and the combined p-value is $p_{\text{Fisher}} = P(\chi^2(2C) \geq T)$. Under $\mathcal{H}_0$, T typically takes values near its mean $2C$. If the observed T is much larger than $2C$, then $p_{\text{Fisher}} = P(\chi^2(2C) \geq T)$ is small, meaning such a large T would be very unlikely without a watermark; hence we reject $\mathcal{H}_0$.

Step 4: Detection strength. From Step 3, $T = 2\sum_c(-\ln p_c)$. To understand how each $-\ln p_c$ depends on k_{eff}, we expand the exact p-value from Step 2. By Step 1, $\tilde{u}_c$ is uniform on $S^{k_{\text{eff}}-1}$. The cosine between a uniform random direction and a fixed direction in k_{eff} dimensions has density $f(s) = C_k\,(1-s^2)^{(k_{\text{eff}}-3)/2}$ on $[-1,1]$ (Mardia & Jupp, 2009), where $C_k = \Gamma(k_{\text{eff}}/2)/[\sqrt{\pi}\,\Gamma((k_{\text{eff}}-1)/2)]$. By Stirling's approximation, $C_k \sim \sqrt{k_{\text{eff}}/(2\pi)}$ for large k_{eff}, so $\ln C_k = \frac{1}{2}\ln k_{\text{eff}} + O(1)$; this justifies the $O(\ln k_{\text{eff}})$ correction term used below. The exponent $(k_{\text{eff}}-3)/2$ determines the shape: for $k_{\text{eff}} \geq 3$ the density is bell-shaped (peaked at zero), enabling the tail approximation below; for $k_{\text{eff}} \leq 2$ (i.e., $C \leq 2$) the density becomes U-shaped and this approximation does not apply, though the exact p-value from Eq. 24 remains valid. The one-sided tail probability is therefore

$$p_c = \int_{s_c}^{1} C_k\,(1-t^2)^{(k_{\text{eff}}-3)/2}\,dt, \quad s_c > 0. \tag{26}$$

To evaluate this integral, we rewrite the integrand in exponential form:

$$(1-t^2)^{(k_{\text{eff}}-3)/2} = \exp\Big(\tfrac{k_{\text{eff}}-3}{2}\ln(1-t^2)\Big). \tag{27}$$

By the exact Taylor expansion $\ln(1-x) = -x - \sum_{n=2}^{\infty} x^n/n$ for $x \in [0,1)$, setting $x = t^2$:

$$(1-t^2)^{(k_{\text{eff}}-3)/2} = \exp\Big(-\tfrac{k_{\text{eff}}}{2}\,t^2\Big)\cdot\exp(-R_1(t)), \tag{28}$$

where $R_1(t) := -\frac{3}{2}\,t^2 + \frac{k_{\text{eff}}-3}{2}\sum_{n=2}^{\infty} t^{2n}/n = \frac{k_{\text{eff}}-3}{2}\big[-\ln(1-t^2)\big] - \frac{k_{\text{eff}}}{2}\,t^2$ is the Taylor remainder, exact for $t \in [0,1)$. Substituting Eq. 28 into Eq. 26:

$$p_c \;=\; C_k \int_{s_c}^{1} \exp\Big(-\tfrac{k_{\text{eff}}}{2}\,t^2\Big)\cdot\exp(-R_1(t))\,dt. \tag{29}$$

The dominant factor $\exp(-\frac{k_{\text{eff}}}{2}t^2)$ is monotonically decreasing for $t > 0$, so the integral is dominated at $t = s_c$ by $\exp(-\frac{k_{\text{eff}}}{2}s_c^2)\cdot\exp(-R_1(s_c))$, with all remaining factors (integration width, C_k, and the residual variation of R_1 across $[s_c, 1]$) contributing only sub-exponentially. Therefore:

$$-\ln p_c \;=\; \tfrac{1}{2}\,k_{\text{eff}}\,s_c^2 \;+\; R_1(s_c) \;+\; \Gamma(k_{\text{eff}}, s_c), \tag{30}$$

where $R_1(s_c)$ is the Taylor remainder from Eq. 28 evaluated at $t = s_c$, and $\Gamma(k_{\text{eff}}, s_c) := -\ln C_k - \ln W(k_{\text{eff}}, s_c)$ is the sub-leading correction, with the effective integration width $W(k_{\text{eff}}, s_c) := \int_{s_c}^{1}\exp\big(-\frac{k_{\text{eff}}}{2}(t^2 - s_c^2) - (R_1(t) - R_1(s_c))\big)\,dt$.

For $s_c \leq 0$, the one-sided p-value has the closed form $p_c^-(s_c, k_{\text{eff}}) := 1 - \frac{1}{2}\,I_{1-s_c^2}\big(\frac{k_{\text{eff}}-1}{2},\,\frac{1}{2}\big) \geq 1/2$, where $I_x(\cdot,\cdot)$ is the regularised incomplete beta function (Proposition 1); hence $-\ln p_c^-(s_c, k_{\text{eff}}) \in [0, \ln 2]$, while the leading term $(k_{\text{eff}}/2)s_c^2$ can reach $k_{\text{eff}}/2$. Their difference $-\ln p_c^-(s_c, k_{\text{eff}}) - (k_{\text{eff}}/2)s_c^2 \in [-k_{\text{eff}}/2,\, \ln 2]$ enters R_{Fisher} (Eq. 32) as an exact bounded correction for each negatively-aligned class. Substituting Eq. 30 into Step 3 (Eq. 25) yields the decomposition of $-\ln p_{\text{Fisher}}$ into a leading term $(k_{\text{eff}}/2)\sum_c s_c^2$ and a bounded residual R_{Fisher}:

$$\boxed{-\ln\,p_{\text{Fisher}} \;=\; \sum_{c=1}^{C}(-\ln p_c) \;=\; \frac{k_{\text{eff}}}{2}\sum_{c=1}^{C} s_c^2 \;+\; R_{\text{Fisher}}(k_{\text{eff}}, \{s_c\})}, \tag{31}$$

where

$$R_{\text{Fisher}}(k_{\text{eff}}, \{s_c\}) \;:=\; \sum_{c:\,s_c>0}\big[R_1(s_c) + \Gamma(k_{\text{eff}}, s_c)\big] \;+\; \sum_{c:\,s_c\leq 0}\big[-\ln p_c^-(s_c, k_{\text{eff}}) - \tfrac{k_{\text{eff}}}{2}s_c^2\big] \tag{32}$$

groups the two sign regimes: for positively aligned classes the per-class Taylor remainder R_1 and sub-leading correction Γ; for negatively aligned classes the closed-form $-\ln p_c^-$ minus the leading-term overshoot. □

Proof of Corollary 1. Throughout this proof, we adopt the empirical Bernstein framework (Maurer & Pontil, 2009) to establish a $(1-\delta_B)$-probability guarantee: for a chosen failure probability $\delta_B \in (0,1)$, the closed-form estimator below gives the mean detection statistic $\mathbb{E}[-\ln p_{\text{Fisher}}]$, and an empirical Bernstein bound on the concentration around this mean yields the $\text{margin}(\delta_B)$ term used later in the proof.

To derive the closed-form estimator of $\mathbb{E}[-\ln p_{\text{Fisher}}]$ in the design parameters (C, k_{eff}, μ), we take expectation of Theorem 1, which requires the mean and variance of the per-class score s_c under $\mathcal{H}_1$:

- **Mean**: $\mathbb{E}[s_c] = \mu$ is the design-target alignment (§3.3, ensured by Algorithm 1's $\mathcal{L}_{\text{align}}$).
- **Variance**: we adopt $\text{Var}(s_c) = 1/k_{\text{eff}}$ under $\mathcal{H}_1$, using Proposition 1's cosine null value as the design-time marginal variance.

With $\mathbb{E}[s_c^2] = \mu^2 + 1/k_{\text{eff}}$ and Theorem 1's leading Stirling correction $\mathbb{E}[R_{\text{Fisher}}] = -C/2$ (relating the per-class Fisher sum $\sum_{c=1}^{C}(-\ln p_c) = T/2$ to $-\ln P(\chi^2_{2C} \geq T)$), taking expectation of Theorem 1 gives:

$$\mathbb{E}[-\ln p_{\text{Fisher}}] \;=\; \tfrac{k_{\text{eff}}}{2} \cdot \underbrace{C \cdot (\mu^2 + 1/k_{\text{eff}})}_{= C\mu^2 + C/k_{\text{eff}}} - \underbrace{C/2}_{\text{Fisher Stirling}} \;=\; \tfrac{C k_{\text{eff}} \mu^2}{2}. \tag{33}$$

The estimator's (Eq. 33) three inputs each have a clear design-time origin: C is the number of classes fixed by the dataset, k_{eff} is the owner-chosen dimension of the shared subspace S (bounded by C, Assumption 1), and μ is the per-class alignment targeted by the injection objective in §3.2, estimated at design time as $\frac{1}{C}\sum_c s_c$ from a small set of calibration suspects that the owner trains on the watermarked dataset $\mathcal{D}_{\text{wm}}$ before its release. With the estimator of Eq. 33 of the mean detection strength in hand, we next compare it to the detection threshold.

With B buyer keys tested in parallel (§3.4), the Bonferroni-corrected significance threshold is τ/B, i.e., $\mathcal{H}_0$ is rejected whenever $p_{\text{Fisher}} < \tau/B$, equivalently the realised log p-value satisfies

$$-\ln p_{\text{Fisher}} \;>\; -\ln(\tau/B) \;=\; \ln(B/\tau). \tag{34}$$

Substituting the estimator (Eq. 33) into Eq. 34 gives the expectation-level detection condition

$$\mathbb{E}[-\ln p_{\text{Fisher}}] \;=\; \tfrac{C\, k_{\text{eff}}\, \mu^2}{2} \;>\; \ln(B/\tau). \tag{35}$$

Eq. 35 bounds $\mathbb{E}[-\ln p_{\text{Fisher}}]$, not the realized statistic that actually crosses the threshold at design time. Bernstein-type inequalities (Boucheron et al., 2013) lower-bound the realized value via $\Pr(T < \mathbb{E}[T] - t) < \delta_B$ using the variance of T; however, $\text{Var}(-\ln p_{\text{Fisher}})$ has no closed form in the design parameters (C, k_{eff}, μ) because it involves the fourth moment of s_c and cross-class covariances induced by the shared GNN backbone, neither of which is analytically expressible. The empirical Bernstein variant (Maurer & Pontil, 2009) bypasses this by replacing the analytical variance with a sample $\hat{\sigma}$ estimated from N calibration suspects. Applied to $-\ln p_{\text{Fisher}}$ around its expectation:

$$\Pr\Big(-\ln p_{\text{Fisher}} < \mathbb{E}[-\ln p_{\text{Fisher}}] - \text{margin}(\delta_B)\Big) < \delta_B, \qquad \text{margin}(\delta_B) := \hat{\sigma} \cdot \sqrt{2\ln(1/\delta_B)}, \tag{36}$$

where $\hat{\sigma}$ is the standard sample standard deviation of $-\ln p_{\text{Fisher}}$. This $\hat{\sigma}$ empirically captures the observed correlation among $\{s_c\}$ directly; Remark 1's compound-symmetric model $\hat{\sigma} = \hat{\sigma}_{\text{indep}} \cdot \sqrt{1 + (C-1)\bar{\rho}}$ (giving $\text{margin}(\delta_B) = \hat{\sigma}_{\text{indep}} \cdot \sqrt{1 + (C-1)\bar{\rho}} \cdot \sqrt{2\ln(1/\delta_B)}$) is provided as an interpretive decomposition of how correlation inflates the margin over the hypothetical independent-classes baseline $\hat{\sigma}_{\text{indep}}$.

Substituting $\mathbb{E}[-\ln p_{\text{Fisher}}] = C k_{\text{eff}} \mu^2/2$ (Eq. 33) into Eq. 36 and requiring the realized $-\ln p_{\text{Fisher}}$ to exceed $\ln(B/\tau)$ with probability $\geq 1 - \delta_B$ yields the *design-time* $(1-\delta_B)$-probability sufficient condition

$$\frac{C\, k_{\text{eff}}\, \mu^2}{2} - \text{margin}(\delta_B) \;>\; \ln(B/\tau), \tag{37}$$

stated purely in the design parameters (C, k_{eff}, μ) and matching Corollary 1's Eq. 2. □

Empirical validation and condition check. Table 8 verifies Corollary 1 on five datasets with $N = 50$ independent suspects per row. *(i) Closed-form accuracy:* the predicted $\text{Estimator} = C\, k_{\text{eff}}\, \mu^2/2$ (Eq. 33) and the observed $\text{Statistic} = -\ln p_{\text{Fisher}}$ track each other closely across all rows, confirming the closed-form estimator. *(ii)* $(1-\delta_B)$*-probability sufficient condition:* at $\delta_B \leq 0.05$ (95% Bernstein confidence), the *tuned* configurations of ogbn-arxiv, Roman-Empire, ogbn-products, and Reddit satisfy $\text{Estimator} - \text{margin}(\delta_B) > \ln(B/\tau) \approx 9.90$ at the marketplace setting $B{=}1000, \tau{=}0.05$. Amazon-Ratings' tuned config reaches only 2%: with $C = 5$, the Bernstein concentration of a sum of only five per-class terms is inherently loose, and the fidelity-compliant canonical config already falls below the threshold (§G.1). Each row also lists the three injection-time knobs $(|V_w|, \theta, \delta)$ that

Table 8: Verification of Corollary 1's $(1-\delta_B)$-probability sufficient condition $\mathrm{Estimator} - \mathrm{margin}(\delta_B) > \ln(B/\tau)$ across five datasets, with $N = 50$ independent suspects per dataset. Estimator $= C\, k_{\mathrm{eff}}\, \mu^2/2$ (Eq. 33); Statistic $= -\ln p_{\mathrm{Fisher}}$ observed (mean $\pm\hat{\sigma}$). The $(1-\delta_B)\%$ column reports the maximum Bernstein confidence at which the condition holds. "Holds?" ✓ marks rows satisfying the condition at $\delta_B \le 0.05$ (95% confidence); × marks rows failing at $\delta_B = 0.05$ but achieving weaker confidence per the $(1-\delta_B)\%$ column; "—" marks rows where Estimator $< \ln(B/\tau)$ (condition fails at every δ_B).

Dataset	Config	$(\lvert V_w\rvert, \theta, \delta)$	C	k_{eff}	μ	Estimator	Statistic	Holds?	$(1-\delta_B)\%$	WDA (%, ↑)	OAA (%, ↑)
ogbn-arxiv	weak	(40, 0.96, 0.5)	40	$24.0_{\pm 1.0}$	$0.087_{\pm 0.025}$	$3.63_{\pm 1.00}$	$4.58_{\pm 1.30}$	×	—	0.0	0.0
	tuned	(3929, 0.96, 0.5)	40	$18.5_{\pm 0.5}$	$0.339_{\pm 0.030}$	$42.65_{\pm 7.17}$	$46.24_{\pm 7.35}$	✓	99	100.0	100.0
Roman-Empire	weak	(1800, 0.20, 0.5)	18	$15.0_{\pm 1.5}$	$0.011_{\pm 0.030}$	$0.02_{\pm 0.03}$	$0.79_{\pm 0.27}$	×	—	0.0	0.0
	tuned	(1800, 0.96, 0.5)	18	$15.6_{\pm 0.5}$	$0.395_{\pm 0.045}$	$22.08_{\pm 4.58}$	$27.49_{\pm 3.67}$	✓	99	100.0	100.0
Amazon-Ratings	weak	(100, 0.50, 0.01)	5	$4.0_{\pm 0.5}$	$0.001_{\pm 0.020}$	$0.00_{\pm 0.10}$	$1.04_{\pm 0.40}$	×	—	0.0	0.0
	canonical	(625, 0.99, 0.5)	5	$4.70_{\pm 0.5}$	$0.853_{\pm 0.123}$	$8.68_{\pm 2.55}$	$10.24_{\pm 1.24}$	×	—	66.0	30.0
	tuned	(5000, 0.99, 0.5)	5	$4.80_{\pm 0.5}$	$0.91_{\pm 0.123}$	$10.16_{\pm 2.55}$	$11.52_{\pm 1.24}$	×	2	98.0	96.0
ogbn-products	weak	(1000, 0.92, 0.01)	47	$13.5_{\pm 2.0}$	$0.070_{\pm 0.020}$	$1.55_{\pm 0.90}$	$2.20_{\pm 0.74}$	×	—	0.0	0.0
	tuned	(15409, 0.92, 0.5)	47	$16.3_{\pm 1.0}$	$0.21_{\pm 0.05}$	$17.80_{\pm 8.48}$	$25.57_{\pm 3.14}$	✓	96	100.0	100.0
Reddit	weak	(500, 0.92, 0.5)	41	$2.0_{\pm 0.0}$	$0.060_{\pm 0.018}$	$0.15_{\pm 0.50}$	$1.10_{\pm 0.38}$	×	—	0.0	0.0
	tuned	(11640, 0.92, 0.5)	41	$29.5_{\pm 5.8}$	$0.177_{\pm 0.023}$	$19.14_{\pm 6.16}$	$21.89_{\pm 2.86}$	✓	99	100.0	100.0

jointly drive μ; *weak* configurations underweight one knob, driving Estimator below threshold and WDA/OAA to zero. Tuned configurations achieve $\mathrm{WDA} \ge 98\%$ empirically, exceeding what the Bernstein margin conservatively predicts. The realized detection concentrates substantially tighter than the theoretical upper bound.

D.6 Rank Bound of the Shared Subspace S

Our detection analysis (Theorem 1 and Corollary 1) is stated in terms of the effective rank k_{eff} of the alignment matrix $\mathbf{X}_{\mathrm{align}} \in \mathbb{R}^{d_1 \times d_2}$, defined for singular values $\sigma_1 \ge \sigma_2 \ge \cdots \ge 0$ and a relative tolerance $\mathrm{tol} \in (0, 1)$ as $k_{\mathrm{eff}} := |\{j : \sigma_j > \mathrm{tol} \cdot \sigma_1\}|$. We motivate two natural boundaries on k_{eff}, then combine them into a single working assumption.

Lower bound $k_{\mathrm{eff}} \ge 3$ (regularity on tol*).* The cosine $s = \langle u, v\rangle$ between two uniformly random unit vectors on $S^{k_{\mathrm{eff}}-1}$ has density $f(s) \propto (1-s^2)^{(k_{\mathrm{eff}}-3)/2}$. This density is bell-shaped only when $k_{\mathrm{eff}} \ge 3$, the regime required by the tail-bound argument in Theorem 1. The condition does not assert anything about $\mathbf{X}_{\mathrm{align}}$'s intrinsic rank; it constrains the choice of tol. At the chosen tolerance, the realized k_{eff} clears 3 on every dataset (Appendix D.5, Table 8), e.g. $k_{\mathrm{eff}} \approx 18.5$ on ogbn-arxiv, with the tightest margin on Amazon-Ratings where k_{eff} approaches the small class count $C = 5$.

Upper bound $k_{\mathrm{eff}} \le C$ (LDA-motivated analogy). Discriminative information for a C-class problem is expected to concentrate in $\lesssim C$ directions of the embedding space, since Fisher's LDA (Fisher, 1936) decomposes such a problem into $C-1$ class-discriminative directions plus one class-mean direction, totalling C. This is a *motivating analogy* rather than a structural derivation: $\mathbf{X}_{\mathrm{align}} = \mathbf{H}_s^{\dagger}\mathbf{H}_\theta$ is the tolerance-thresholded numerical rank of a least-squares fit between two independently trained networks' embedding spaces, not a between-class scatter matrix. As such, it can in principle retain non-negligible singular values above C from optimization noise, architecture-specific structure, or the free choice of tol. A structural derivation would require an explicit low-rank assumption connecting $\mathbf{X}_{\mathrm{align}}$'s rank to the class structure together with control of finite-sample noise and tol. In lieu of such a derivation, the bound is checked per dataset in Appendix D.5, Table 8: e.g. $k_{\mathrm{eff}} \approx 18.5 < C = 40$ on ogbn-arxiv, and all remaining datasets similarly satisfy $k_{\mathrm{eff}} \le C$.

Combining the two boundaries, we formalize the working assumption used throughout the detection analysis.

Assumption 1 (Effective rank of the alignment matrix)**.** *Let C be the number of classes, and let $\mathbf{X}_{\mathrm{align}} \in \mathbb{R}^{d_1 \times d_2}$ be an alignment matrix between the embedding spaces of two C-class classifiers of dimensions d_1 and d_2, with singular values $\sigma_1 \ge \sigma_2 \ge \cdots \ge 0$. For a relative tolerance* $\mathrm{tol} \in (0, 1)$*, define the effective rank $k_{\mathrm{eff}} := \big|\{j : \sigma_j > \mathrm{tol} \cdot \sigma_1\}\big|$. Throughout our detection analysis we adopt*

$$3 \;\le\; k_{\mathrm{eff}} \;\le\; C, \tag{38}$$

where the lower bound is a regularity condition on tol *(needed for the cosine-density tail argument in Theorem 1), and the upper bound is motivated by an LDA analogy on C-class problems.*

D.7 Owner Attribution: Proof of Corollary 2 and Tightening Lemmas

Proof outline. The derivation proceeds in four steps: the per-class mean of the foreign cosine s_c; the i.i.d. angular factorisation that collapses the product into $(1-\bar{P}^\star)^{B-1}$ via the closed-form $P^\star(\varphi) = \exp(-T^\star \sin^2\varphi)$ obtained by matching two distributional endpoints; the detection precondition $\mathbf{D}$ in Eq. 42, which is used as the indicator $\mathbf{1}[\mathbf{D}]$ in the lower bound on attribution success of Eq. 3; and the consistency-penalised statistic of Eq. 50 that tightens the approximation at small k_{eff}.

Per-class mean (own and foreign). For the own key, $\mu := \mathbb{E}[s_c \mid \text{own}] = \mathbb{E}[\cos(\tilde{\mathbf{W}}_c, \tilde{u}_c)] \in (0, 1]$ is the post-training expected cosine alignment between the suspect's whitened classifier weight $\tilde{\mathbf{W}}_c$ and the own carrier $\tilde{u}_c$ (§3.3); $\mu > 0$ is ensured by Algorithm 1's alignment loss $\mathcal{L}_{\text{align}}$. For a foreign key at angle φ from own, decomposing $u_{\text{foreign}} = \cos\varphi\, u_{\text{own}} + \sin\varphi\, w$ with w uniform on the perpendicular sphere yields, by rotation invariance of the uniform distribution,

$$\mathbb{E}[s_c \mid \varphi] \;=\; \mu\cos\varphi. \tag{39}$$

Both own and foreign share $\text{Var}(s_c) = 1/k_{\text{eff}}$ under $\mathcal{H}_0$ (Proposition 1).

Conditional-independence factorisation. Define the *per-pair foreign-winning probability*

$$P^\star(\varphi) \;:=\; \Pr\big(p^{(\text{foreign})}_{\text{Fisher}} < p^{(\text{own})}_{\text{Fisher}} \;\big|\; \varphi\big), \tag{40}$$

where $\varphi := \angle(\tilde{u}^{\text{foreign}}, \tilde{u}^{\text{own}})$ is the angle between the foreign and own carrier directions on $\mathbb{S}^{k_{\text{eff}}-1}$, so $P^\star(\varphi)$ is the probability that a single foreign at angle φ beats own's Fisher p-value. Its closed form $P^\star(\varphi) \approx \exp(-T^\star \sin^2\varphi)$ is derived below by matching distributional endpoints (Eq. 45). Conditional on own's typical-case detection statistic $T^\star := -\ln p^{(\text{own})}_{\text{Fisher}}$ and the pool angles $\{\varphi_j\}$, the foreigns' win events are independent because the $B-1$ foreign carriers are sampled independently. Each individual win event has probability $P^\star(\varphi_j)$, so under the following assumption the joint loss probability factorises: *Assumption (angle-sufficient factorisation).* We treat the angle φ_j as the sole summary of the shared dependencies among foreign win events (e.g., the suspect's classifier weights $\mathbf{W}_c$, training data distribution, overfitting patterns). Under this working assumption:

$$\Pr\big(p^{(\text{own})}_{\text{Fisher}} < \min_{j\neq\text{own}} p^{(j)}_{\text{Fisher}} \;\big|\; T^\star, \{\varphi_j\}\big) \;=\; \prod_{j\neq\text{own}} \big(1 - P^\star(\varphi_j)\big). \tag{41}$$

We restrict attention to $\varphi \in (0, \pi/2]$: foreigns at $\varphi > \pi/2$ have $\mathbb{E}[s_c \mid \varphi] = \mu\cos\varphi < 0$, so most s_c are absorbed into the residual by Theorem 1 assumption (iii) and pose no threat.

Detection precondition $\mathbf{D}$. For Eq. 41 to translate into a guarantee on $\hat{i} = \arg\min_i p^{(i)}_{\text{Fisher}}$, own must itself fire detection. The raw firing condition $-\ln p^{(\text{own})}_{\text{Fisher}} > \ln(1/\tau)$ depends on the random per-class scores $\{s_c\}$ and is not analytically tractable; we replace it with a closed-form sufficient condition in $(C, k_{\text{eff}}, \mu, \tau)$. Theorem 1 gives $-\ln p_{\text{Fisher}} = (k_{\text{eff}}/2)\sum_c s_c^2 + R$ with $R = O(\ln k_{\text{eff}})$. Taking the worst-case residual $R = -\ln k_{\text{eff}}$ yields the safe lower bound $-\ln p_{\text{Fisher}} \geq (k_{\text{eff}}/2)\sum_c s_c^2 - \ln k_{\text{eff}}$. Substituting the closed-form estimator $\mathbb{E}[(k_{\text{eff}}/2)\sum_c s_c^2] = Ck_{\text{eff}}\mu^2/2$ (Eq. 33) and demanding the lower bound clears the threshold gives:

$$\mathbf{D}: \quad Ck_{\text{eff}}\,\mu^2 \;\geq\; 2\ln(k_{\text{eff}}/\tau). \tag{42}$$

This precondition matches the structural form of Corollary 1, Eq. 2, with the worst-case residual term $\ln(k_{\text{eff}}/\tau)$ in place of the Bonferroni-corrected $\ln(B/\tau)$. $\mathbf{D}$ in Eq. 42 is a *sufficient* condition for actual detection firing: $\mathbf{D} \Rightarrow \{-\ln p^{(\text{own})}_{\text{Fisher}} > \ln(1/\tau)\}$ via the worst-case residual $R = -\ln k_{\text{eff}}$ in Theorem 1. As an indicator-set inclusion, this means

$$\mathbf{1}[\mathbf{D}] \;\leq\; \mathbf{1}[\text{actual firing}].$$

Now decompose the joint success event conditional on $(T^\star, \{\varphi_j\})$. Writing $\Pr_c(\cdot) := \Pr(\cdot \mid T^\star, \{\varphi_j\})$ for brevity:

$$\begin{aligned}\Pr\big(\text{detection \& attribution success}\big) &= \mathbf{1}[\text{actual firing}] \cdot \Pr_c\big(p^{(\text{own})}_{\text{Fisher}} < \min_j p^{(j)}_{\text{Fisher}}\big) \\ &= \mathbf{1}[\text{actual firing}] \cdot \prod_{j \neq \text{own}} (1 - P^\star(\varphi_j)) \quad \text{(Eq. 41)} \\ &\geq \mathbf{1}[\mathbf{D}] \cdot \prod_{j \neq \text{own}} (1 - P^\star(\varphi_j))\end{aligned}$$

The third line replaces $\mathbf{1}[\text{actual firing}]$ with the smaller indicator $\mathbf{1}[\mathbf{D}]$, weakening the right-hand side and giving the $\geq$. Marginalising over the conditioning yields the joint guarantee:

$$\Pr\big(\text{detection \& attribution success}\big) \;\geq\; \mathbf{1}[\mathbf{D}] \cdot \prod_{j \neq \text{own}} \big(1 - P^\star(\varphi_j)\big). \tag{43}$$

The right-hand side still depends on the per-foreign angles $\{\varphi_j\}$. Two further transformations remove this dependence: **Step (i)** averages the product over the random pool to collapse $\prod_j (1 - P^\star(\varphi_j))$ into a single scalar $(1 - \bar{P}^\star)^{B-1}$; **Step (ii)** replaces $P^\star(\varphi)$ with a closed-form expression in $T^\star$ and φ to remove the angle dependence inside $\bar{P}^\star$.

Step (i): Factorisation via independent angular sampling. Treating the $B-1$ foreign carriers as independent and identically distributed draws uniform on $\mathbb{S}^{k_{\text{eff}}-1}$, with a fresh *pool* re-sampled per realisation as in Section 3.4, the independence of the angles φ_j turns the product into the exact factorisation

$$\mathbb{E}_{\text{pool}}\left[\prod_{j \neq \text{own}} \big(1 - P^\star(\varphi_j)\big)\right] \;=\; \big(1 - \bar{P}^\star\big)^{B-1}, \qquad \bar{P}^\star \;:=\; \mathbb{E}_\varphi\big[P^\star(\varphi)\big], \tag{44}$$

where $\mathbb{E}_{\text{pool}}$ averages over the random sampling of the $B-1$ foreign carriers, and $\mathbb{E}_\varphi$ averages a single foreign's angle φ under the angular density $f(\varphi)$ on the sphere (defined explicitly in Eq. 47 below). Substituting Eq. 44 into Eq. 43 reduces the product to a scalar:

$$\Pr\big(\text{detection \& attribution success}\big) \;\geq\; \mathbf{1}[\mathbf{D}] \cdot \big(1 - \bar{P}^\star\big)^{B-1}.$$

What remains is to express $\bar{P}^\star = \mathbb{E}_\varphi[P^\star(\varphi)]$ in closed form.

Step (ii): Closed-form for $P^\star(\varphi)$. We use Theorem 1's closed-form estimator. Let $T^\star := -\ln p^{(\text{own})}_{\text{Fisher}}$ denote the measured detection statistic on own (so $T^\star = C k_{\text{eff}} \mu^2/2$ by Eq. 33). We pin down the distribution of $-\ln p_{\text{foreign}}$ at angle φ by matching its two known endpoints.

Endpoint 1 (mean at general φ). By Eq. 39, a foreign at angle φ has per-class cosine $\mathbb{E}[s_c \mid \varphi] = \mu \cos\varphi$, so $\sum_c s_c^2 = C\mu^2 \cos^2\varphi$, and Theorem 1 gives

$$\mathbb{E}[-\ln p_{\text{foreign}} \mid \varphi] \;=\; \tfrac{k_{\text{eff}}}{2} C\mu^2 \cos^2\varphi \;=\; T^\star \cos^2\varphi.$$

Endpoint 2 (full distribution at $\varphi = \pi/2$). The signal vanishes ($\mu\cos(\pi/2) = 0$), so foreign is at $\mathcal{H}_0$. Any well-defined p-value is uniformly distributed under its null (probability integral transform), so $p_{\text{Fisher}} \sim \text{Uniform}[0,1]$ *exactly*. Then $-\ln p_{\text{foreign}} \sim \text{Exp}(1)$ by change of variable: for $U \sim \text{Uniform}[0,1]$, $\Pr(-\ln U > x) = \Pr(U < e^{-x}) = e^{-x}$ (unit-rate exponential).

Additive signal-plus-noise model. The simplest distribution that matches both endpoints decomposes the foreign log-evidence into a φ-dependent deterministic signal plus an $\text{Exp}(1)$ noise term:

$$-\ln p_{\text{foreign}} \;\approx\; \underbrace{T^\star \cos^2\varphi}_{\text{signal (Endpoint 1)}} \;+\; \underbrace{\varepsilon}_{\text{noise (Endpoint 2)}}, \qquad \varepsilon \sim \text{Exp}(1).$$

Consistency with both endpoints. At $\varphi = \pi/2$, the signal vanishes ($\cos^2(\pi/2) = 0$), giving $-\ln p_{\text{foreign}} = \varepsilon \sim \text{Exp}(1)$, which *exactly* matches Endpoint 2. At general φ, the mean is

$\mathbb{E}[-\ln p_{\text{foreign}}] = T^\star \cos^2\varphi + \mathbb{E}[\varepsilon] = T^\star \cos^2\varphi + 1$, matching Endpoint 1 up to a constant offset $1 \ll T^\star$. Plugging into $P^\star(\varphi)$:

$$\begin{aligned} P^\star(\varphi) \;&=\; \Pr\big(-\ln p_{\text{foreign}} > T^\star\big) \;=\; \Pr\big(T^\star\cos^2\varphi + \varepsilon > T^\star\big) \\ &= \Pr\big(\varepsilon > T^\star(1-\cos^2\varphi)\big) \;=\; \Pr\big(\varepsilon > T^\star \sin^2\varphi\big) \\ &= \exp(-T^\star \sin^2\varphi), \end{aligned}$$

where the last step plugs $x = T^\star \sin^2\varphi$ into the $\mathrm{Exp}(1)$ tail $\Pr(\varepsilon > x) = e^{-x}$ derived in Endpoint 2. We thus obtain the closed-form per-pair foreign-winning probability

$$P^\star(\varphi) \;\approx\; \exp(-T^\star \sin^2\varphi). \tag{45}$$

The form is exact at $\varphi = \pi/2$ and an interpolation elsewhere. Substituting this closed-form $P^\star(\varphi)$ into the factorisation Eq. 44 completes the proof of Corollary 2, restated here for reference:

Corollary (Cor. 2, Top-1 Multi-Buyer Attribution, restated)**.**

$$\boxed{\Pr\big(\textit{detection \& attribution success}\big) \;\approx\; \mathbf{1}[\mathbf{D}] \cdot \big(1 - \bar{P}^\star\big)^{B-1}} \tag{46}$$

The boxed inequality leaves one quantity unspecified, $\bar{P}^\star = \mathbb{E}_\varphi[P^\star(\varphi)]$, which requires the distribution of φ. For a foreign carrier sampled uniformly on $\mathbb{S}^{k_{\text{eff}}-1}$, the angle to a fixed direction follows the standard spherical density (Iscen et al., 2018)

$$f(\varphi) \;=\; \frac{1}{Z}\sin^{k_{\text{eff}}-2}(\varphi), \qquad Z = \int_0^{\pi/2} \sin^{k_{\text{eff}}-2}(\varphi)\,\mathrm{d}\varphi, \qquad \varphi \in [0, \pi/2], \tag{47}$$

equivalent under $y = \sin^2\varphi$ to $\cos^2\varphi \sim \mathrm{Beta}(1/2, (k_{\text{eff}}-1)/2)$; this is the Beta-CDF form used for cosine p-value computation in dataset watermarking (Sablayrolles et al., 2020). Substituting Eq. 45 into the expectation and integrating against $f(\varphi)$:

$$\bar{P}^\star \;=\; \int_0^{\pi/2} \exp(-T^\star \sin^2\varphi)\, f(\varphi)\,\mathrm{d}\varphi. \tag{48}$$

Eq. 48 is evaluated by one-dimensional numerical quadrature (no further approximation). The domain $[0, \pi/2]$ excludes foreigns at $\varphi > \pi/2$, which have $\mathbb{E}[s_c \mid \varphi] < 0$ and do not fire detection; the only remaining approximation in the entire chain is Theorem 1's closed-form estimator inside $P^\star(\varphi)$. □

Variance bound. The approximation from Corollary 2 (Eq. 46) pins down only the mean of the success outcome; to quantify how tightly the empirical attribution rate concentrates around this mean we additionally derive a closed-form estimate of $\mathrm{Var}(I)$, reported in the last column of Table 9. The success indicator $I := \mathbf{1}\{\text{detection \& attribution success}\}$ is the AND of two binary factors: own's detection and own's argmin victory:

$$I \;=\; \underbrace{\mathbf{1}[\mathbf{D}]}_{\text{own fires detection}} \cdot \underbrace{\prod_{j \neq \text{own}} \mathbf{1}\big\{p^{(\text{own})}_{\text{Fisher}} < p^{(j)}_{\text{Fisher}}\big\}}_{\text{own beats every foreign}}.$$

The product of binary factors is itself binary, so $I \in \{0, 1\}$ and $I^2 = I$ regardless of the internal multi-key structure. This yields the standard Bernoulli identity:

$$\mathrm{Var}(I) \;=\; \mathbb{E}[I^2] - (\mathbb{E}[I])^2 \;=\; p - p^2 \;=\; p(1-p), \qquad p := \mathbb{E}[I] = \Pr(I = 1).$$

The angular density $f(\varphi)$ enters $\mathrm{Var}(I)$ *only through* p: marginalising the pool $\{\varphi_j\} \sim f(\varphi)$ (i.i.d.) and the per-foreign Bernoulli outcomes via Eq. 44 yields $p \approx L := \mathbf{1}[\mathbf{D}] \cdot (1 - \bar{P}^\star)^{B-1}$ from Eq. 46, with $\bar{P}^\star$ integrating $f(\varphi)$ via Eq. 48. Combined with concavity of $p(1-p)$ on $[0, 1]$ (maximised at $p = 1/2$), this gives the closed-form estimate:

$$\mathrm{Var}(I) \;\approx\; \begin{cases} L\,(1-L), & L \geq 1/2 \\ 1/4, & L < 1/2 \end{cases}. \tag{49}$$

Tighter L (closer to 1) implies smaller $\mathrm{Var}(I)$: at $L = 1$, $\mathrm{Var}(I) = 0$ (deterministic success); at $L = 1/2$, the bound is $1/4$ (maximum Bernoulli variance).

Table 9: Approximate joint detection-and-attribution success probability for a single watermarked suspect at $B=1000$, $\tau=0.05$, with each dataset's parameters tuned to satisfy Corollary 1 so that $\mathbf{1}[\mathbf{D}]=1$, aggregated over 50 watermarked suspects per dataset. $B_{\max}$ is the largest buyer count keeping Empirical Calibration 1's approximate rate at or above 0.95, computed as $\lfloor \ln(0.95)/\ln(1-\bar{P}^{\star}_{\text{pen}}) \rfloor + 1$. Column header **Emp. Cal. 1** = Empirical Calibration 1.

	Cor. 1 LHS components	Cor. 2 RHS components			Approximate attribution success probability (↑)		Maximum buyer count	Variance bound (↓)	Owner Attribution Accuracy (%, ↑)	
Dataset	$(\lvert V_w\rvert, \theta, \delta)$	$\mathbf{1}[\mathbf{D}]$	B	$\bar{P}^{\star}$	Cor. 2 (Eq. 46)	Emp. Cal. 1 (Eq. 52)	$B_{\max}$ @ Emp. Cal. 1 approx. rate ≥ 0.95	$\mathrm{Var}(I)$ (Eq. 49)	$\arg\min p_{\text{Fisher}}$ (Cor. 2)	$\arg\min T_i$ (Emp. Cal. 1, Eq. 52)
ogbn-arxiv	(3929, 0.96, 0.5)	1	1000	2.4×10^{-10}	1.000	$1.000_{\pm 0.000}$	$\leq 2.1\times10^{8}$	2.4×10^{-7}	100.0	100.0
Roman-Empire	(1800, 0.96, 0.5)	1	1000	3.7×10^{-7}	1.000	$1.000_{\pm 0.001}$	$\leq 1.4\times10^{5}$	3.7×10^{-4}	100.0	100.0
ogbn-products	(15409, 0.92, 0.5)	1	1000	1.4×10^{-6}	0.999	$0.997_{\pm 0.001}$	$\leq 3.5\times10^{4}$	1.4×10^{-3}	100.0	100.0
Reddit	(11640, 0.92, 0.5)	1	1000	2.8×10^{-8}	1.000	$1.000_{\pm 0.000}$	$\leq 1.8\times10^{6}$	2.8×10^{-5}	100.0	100.0
Amazon-Ratings	(625, 0.99, 0.5)	0	1000	1.4×10^{-2}	0	0	—	0	24.0	30.0
	(5000, 0.99, 0.5)	1	1000	1.0×10^{-2}	4.0×10^{-5}	$0.205_{\pm 0.300}$	$\leq 7.5\times10^{3}$	7.0×10^{-3}	96.0	98.0

Empirical Calibration 1: penalty-shifted $\bar{P}^{\star}_{\text{pen}}$. Corollary 2's approximation is tight at high k_{eff} but loose at small k_{eff}, where $\bar{P}^{\star}$ remains large. Empirical Calibration 1 tightens it by exploiting the consistency asymmetry between own and foreign keys, shrinking $\bar{P}^{\star} \to \bar{P}^{\star}_{\text{pen}}$. Because at small k_{eff} a randomly sampled foreign carrier can by chance align strongly with a single class direction, producing one large s_c while the rest stay small in a "spiky" pattern that lets Fisher's combination beat own's p_{Fisher} despite poor overall alignment, we tighten Cor. 2 with a consistency-penalised statistic

$$T_i = \log_{10} p^{(i)}_{\text{Fisher}} + \lambda \cdot \hat{\sigma}\big(\{s_c^{(i)}\}\big), \qquad i^{\star} = \arg\min_i T_i, \tag{50}$$

where $\hat{\sigma}$ is the standard deviation of $\{s_c\}$, small for own which is uniform across classes and large for spiky foreigns, and $i^{\star}$ denotes the resulting attribution decision under the penalised statistic. The penalty raises foreigns' required overshoot by $\lambda \ln 10 \cdot \Delta\hat{\sigma}$, shrinking $\bar{P}^{\star} \to \bar{P}^{\star}_{\text{pen}}$, with

$$\Delta\hat{\sigma} \;:=\; \mathbb{E}[\hat{\sigma}_{\text{foreign}} - \hat{\sigma}_{\text{own}}] \;\approx\; \frac{1}{N}\sum_{n=1}^{N}\big(\hat{\sigma}_{\text{foreign},n} - \hat{\sigma}_{\text{own},n}\big) \tag{51}$$

is the gap between the foreign and own per-class standard deviation of $\{s_c\}$. Substituting $\bar{P}^{\star}_{\text{pen}} := 10^{-\lambda\Delta\hat{\sigma}}\bar{P}^{\star}$ into Eq. 46:

$$\boxed{\Pr(\text{detection \& attribution success} \mid \text{penalty}) \;\approx\; \mathbf{1}[\mathbf{D}] \cdot \big(1 - \bar{P}^{\star}_{\text{pen}}\big)^{B-1}} \tag{52}$$

This is the semi-empirical penalty-shifted sister of Eq. 46. We calibrate $\Delta\hat{\sigma}$ once per dataset by averaging $\hat{\sigma}_{\text{foreign},n} - \hat{\sigma}_{\text{own},n}$ over 50 watermarked suspects, each contrasted with 999 foreign carriers, totaling $N=9990$ pairs. Per-suspect $\Delta\hat{\sigma}$ variability propagates to Empirical Calibration 1's approximate rate (Table 9): stable at ± 0.001 or smaller for ogbn-arxiv, Roman-Empire, ogbn-products, and Reddit, but ± 0.300 for Amazon-Ratings due to $C=5$ concentration limits.

Per-dataset numerical evaluation. We numerically evaluate Corollary 2's closed-form approximation on five datasets to confirm that it predicts high attribution success across the marketplace; Empirical Calibration 1 provides a refinement that tightens the approximation in regimes where Corollary 2 is conservative, and the empirical OAA confirms the predicted rates translate into practical success.

Table 9 reports the approximate rates at $B=1000$. **Corollary 2**'s closed-form approximation saturates on the four non-Amazon datasets, where the sustainable buyer count $B_{\max}$, defined as the largest B keeping Empirical Calibration 1's approximate rate at 0.95 or above, ranges from 3.5×10^4 on ogbn-products to 2.1×10^8 on ogbn-arxiv, far exceeding the marketplace size of 1000. Amazon-Ratings appears in two rows for two values of $|V_w|$, exposing two limitations of the approximation at low effective rank. Under the fidelity-bounded $|V_w|=625$, Corollary 1's sufficient condition fails so $\mathbf{1}[\mathbf{D}]=0$ and both approximations collapse to zero, yet **Empirical Calibration 1**'s consistency penalty lifts the empirical OAA from 24% to 30%. Under $|V_w|=5000$, Corollary 1 holds yet the low effective rank keeps $\bar{P}^{\star} \approx 10^{-2}$, so the raw Corollary 2 approximation is only 4×10^{-5}; Empirical Calibration 1's per-suspect approximate rate is 0.205 ± 0.300 across the 50 calibration suspects, reflecting the high per-suspect variability of $\Delta\hat{\sigma}$ at $C=5$; Corollary 2's $\arg\min p_{\text{Fisher}}$ rule reaches empirical OAA 96%, and Empirical Calibration 1's $\arg\min T_i$ rule reaches 98%, because marketplace attribution effectively averages the correction over the whole pool.

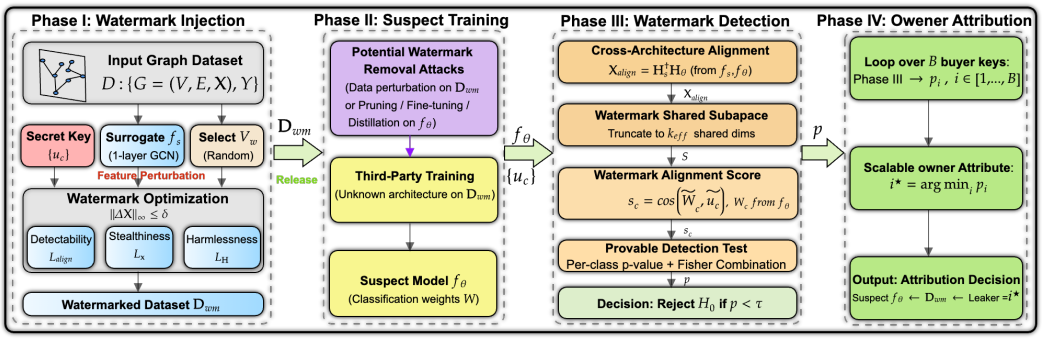


Figure 6: **End-to-end pipeline.** Phases I/III/IV are owner-side (injection, detection, attribution); Phase II is the buyer's suspect training on $\mathcal{D}_{\text{wm}}$, opaque to the owner.

E Method Design Details

E.1 Algorithm Pseudo-code

We provide the full pseudo-code for our three algorithms: Algorithm 1 (watermark injection), Algorithm 2 (single-key watermark detection), and Algorithm 3 (multi-buyer attribution; §3.4). Figure 6 provides the end-to-end 4-phase pipeline overview that these algorithms instantiate, complementing the geometric intuition in Figure 2 (main paper).

Algorithm 1 Watermark Injection

Require: Graph $G=(V,E,\mathbf{X})$, labels Y, C classes, perturbation bound δ, target cosine θ
Ensure: Watermarked features $\mathcal{D}_{\text{wm}}$, carriers $\{u_c\}_{c=1}^{C}$, nodes V_w

Train surrogate f_s (1-layer linear GCN) on (G,Y)
Generate $u_c \sim \text{Unif}(\mathbb{S}^{d-1})$, $c=1,\dots,C$
Select V_w: class-balanced random sampling
Initialise $\Delta\mathbf{x}_v \leftarrow \mathbf{0}$, $\forall v \in V_w$
for $t=1,\dots,T$ **do**
  $\Delta\mathbf{H}_v \leftarrow \mathbf{H}_s(\mathbf{x}_v+\Delta\mathbf{x}_v) - \mathbf{H}_s(\mathbf{x}_v)$
  Compute $\mathcal{L}$ (Eq. 1); update $\Delta\mathbf{x}_v$
  $\Delta\mathbf{x}_v \leftarrow \text{clamp}(\Delta\mathbf{x}_v, -\delta, +\delta)$ {L_∞ proj.}
  if $\frac{1}{C}\sum_{c=1}^{C}\cos(\Delta\mathbf{H}_v, u_c) \ge \theta$ **then**
    break
  end if
end for
$\mathbf{X}_{\text{wm}}[V_w] \leftarrow \mathbf{X}[V_w] + \Delta\mathbf{X}$
$\mathcal{D}_{\text{wm}} \leftarrow (V,E,\mathbf{X}_{\text{wm}},Y)$
return $\mathcal{D}_{\text{wm}}, \{u_c\}_{c=1}^{C}, V_w$

Algorithm 2 Watermark Detection

Require: Suspect model f_θ, surrogate f_s, carriers $\{u_c\}_{c=1}^{C}$, hold-out nodes, tol, τ
Ensure: Decision: reject $\mathcal{H}_0$ (watermarked) or not

Extract $\mathbf{H}_s$, $\mathbf{H}_\theta$ on hold-out nodes
$\mathbf{X}_{\text{align}} \leftarrow \mathbf{H}_s^\dagger \mathbf{H}_\theta$ {least-squares alignment}
SVD: $\mathbf{X}_{\text{align}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$
$k_{\text{eff}} \leftarrow |\{j : \sigma_j > \text{tol}\cdot\sigma_1\}|$ {$\le C$, Asm. 1}
$\mathbf{M} \leftarrow \mathbf{V}_k\mathbf{\Sigma}_k^{-1}\mathbf{V}_k^\top$ {whiten to k_{eff}-dim subspace}
for $c=1,\dots,C$ **do**
  $\tilde{u}_c, \tilde{\mathbf{W}}_c$ via §3.3
  $s_c \leftarrow \tilde{u}_c \cdot \tilde{\mathbf{W}}_c$ {alignment score}
  p_c via Prop. 1 with $d = k_{\text{eff}}$
end for
p_{Fisher} via Fisher's method {Thm. 1}
return reject $\mathcal{H}_0$ if $p_{\text{Fisher}} < \tau$

Algorithm 3 Multi-Buyer Attribution

Require: Suspect f_θ, surrogate f_s, B buyer carrier sets $\{\{u_c^{(i)}\}_{c=1}^{C}\}_{i=1}^{B}$, hold-out nodes, tol, τ
Ensure: Attribution decision: identified leaker $i^\star \in [B]$ with dataset $\mathcal{D}_{\text{wm}}^{(i^\star)}$, or "no detection"

for $i=1,\dots,B$ **do**
  $p_{\text{Fisher}}^{(i)} \leftarrow$ Algorithm 2 with carriers $\{u_c^{(i)}\}_{c=1}^{C}$
end for
if $\min_i p_{\text{Fisher}}^{(i)} \ge \tau$ **then**
  return "no detection" {no buyer's key fires}
end if
$i^\star \leftarrow \arg\min_{i\in[B]} p_{\text{Fisher}}^{(i)}$ {Cor. 2's top-1 rule}
return $i^\star$ {f_θ traced to buyer $i^\star$, dataset $\mathcal{D}_{\text{wm}}^{(i^\star)}$}

Table 10: Watermark node selection strategy comparison on ogbn-arxiv and GCN.

	Fidelity	**Detection**	**Harmfulness**		**Stealthiness**		
Strategy	$\Delta\mathrm{Acc}\,(\%)$	$\log_{10}(p)\downarrow$	$H\downarrow$	$\hat{H}\downarrow$	$\mathcal{S}\downarrow$	$\bar{\mathcal{S}}\downarrow$	$\mathcal{S}_{\mathrm{knn}}\downarrow$
Random sampling	-0.62	-42.83	0.34	-0.320	0.50	1.46	$\mathbf{1.79}$
Graph distance	$\mathbf{-0.48}$	-46.25	0.30	-0.356	0.50	1.48	$\mathbf{1.79}$
Node degree	-0.71	$\mathbf{-50.33}$	0.24	-0.408	0.50	1.49	1.80
Node classifier confidence	-0.53	-26.21	$\mathbf{0.12}$	-0.146	0.50	$\mathbf{1.38}$	1.81
Node embedding diversity	-0.73	-46.04	0.33	$\mathbf{-0.414}$	0.50	1.47	1.93

Table 11: Optimization-landscape condition numbers and SGD/L-BFGS reachable alignment for $K=1$ vs $K=2$ linear GCN surrogates on ogbn-arxiv.

Metric	$K=1$ ($\mathbf{J}=\mathbf{W}_1$)	$K=2$ ($\mathbf{J}=\mathbf{W}_2\mathbf{W}_1$)
Jacobian condition number $\kappa(\mathbf{J})$	2.9×10^{3}	4.3×10^{6}
Hessian condition number $\kappa(\mathbf{H})=\kappa(\mathbf{J})^2$	8.6×10^{6}	1.8×10^{13}
Max $\cos(\Delta\mathbf{H}_v, u_c)$ with SGD	$\mathbf{1.0}$	0.80
Max $\cos(\Delta\mathbf{H}_v, u_c)$ with L-BFGS	$\mathbf{1.0}$	0.98

E.2 Watermark Node Selection Strategies

Beyond class-balanced random sampling (Section 3.2), we consider four per-class selection strategies exploiting graph structure or model information. **(i) Graph distance (node separation):** pick watermark nodes at graph distance ≥ 2 so no two are direct neighbours, which reduces interference between watermark nodes' embedding shifts during GNN message passing. **(ii) Node degree (low-degree):** pick nodes with few neighbours. Since GNN aggregation averages over node degree, a small feature change on a low-degree node produces a larger embedding shift, so each unit of perturbation budget yields a stronger watermark signal. **(iii) Node classifier confidence (high-confidence):** pick nodes the surrogate classifies with high confidence. Since they are far from any decision boundary, the same perturbation budget is less likely to flip their predicted class, so the watermark causes less harm to downstream task accuracy. **(iv) Node embedding diversity (embedding-space coverage):** pick nodes greedily, taking at each step the one whose embedding is most different from those already chosen. Since the gradient on $\mathbf{W}_c$ is a linear combination of node embeddings, similar embeddings produce redundant collinear contributions confined to a low-dimensional subspace, while diverse embeddings produce independent contributions spanning multiple basis directions, so $\mathbf{W}_c$ aligns more strongly with u_c.

Table 10 evaluates the five strategies on ogbn-arxiv. *Random sampling* already achieves strong detection ($\log_{10} p = -42.83$), with the four heuristics offering different tradeoffs. *Graph distance* (i) slightly improves detection (-46.25), consistent with reduced interference between watermark nodes' embedding shifts. *Node degree* (ii) achieves the strongest detection (-50.33), since low-degree nodes amplify each perturbation's embedding shift through GCN aggregation, but this amplification also propagates to neighboring benign nodes, incurring a slight fidelity drop ($\Delta\mathrm{Acc} = -0.71$). *Node classifier confidence* (iii) trades detection (-26.21) for the lowest harmfulness ($H = 0.12$), since high-confidence nodes resist class flipping under perturbation. *Node embedding diversity* (iv) provides modest gain (-46.04).

E.3 Surrogate Design

Surrogate Design. *(i) Shallower is Better.* For a linear K-layer GCN, the Jacobian $\mathbf{J} = \partial\Delta\mathbf{H}_v/\partial\Delta\mathbf{x}_v$ of the cosine-alignment loss $\mathcal{L}_{\mathrm{align}}$ reduces to $\mathbf{W}_K\cdots\mathbf{W}_1$, so the Hessian of $\mathcal{L}_{\mathrm{align}}$, $\mathbf{H}=\mathbf{J}^\top\mathbf{J}$, has condition number $\kappa(\mathbf{H})=\kappa(\mathbf{J})^2$ that blows up multiplicatively with depth. Table 11 shows that $\kappa(\mathbf{H}) = 8.6\times10^6$ at $K=1$, within SGD's effective range, versus 1.8×10^{13} at $K=2$, requiring expensive L-BFGS. Consequently, SGD reaches $\cos(\Delta\mathbf{H}_v, u_c)\approx1.0$ at $K=1$ but caps at $\cos\approx0.8$ at $K=2$. Table 12 shows that 1-layer linear surrogates achieve much stronger detection than deeper ones.

(ii) Linearity. Non-linear activations (ReLU, Leaky ReLU) produce non-smooth loss landscapes that prevent SGD from reaching the target alignment θ, so we stick to linear surrogates. Table 13 shows that linear surrogate outperforms non-linear surrogates.

Table 12: Effect of surrogate depth K on two datasets (surrogate and suspect both GCN; suspect depth $N=2$, $\theta=0.96$). $K=1$ (linear) outperforms deeper surrogates. $K=0$ is excluded by design (no graph aggregation).

		Fidelity	**Stealthiness**		**Harmfulness**		**Detectability** $\log_{10}(p)$	
Dataset	Surrogate K	ΔAcc (%)	$\mathcal{S}\downarrow$	$\bar{\mathcal{S}}\downarrow$	$H\downarrow$	$\hat{H}\downarrow$	Benign $\uparrow$	WM $\downarrow$
ogbn-arxiv $C=40$	1	$+0.25$	0.50	1.43	0.49	-0.159	-0.42	$\mathbf{-35.49}$
	2	$+1.49$	0.50	1.06	0.48	-0.171	-0.41	-11.01
	3	$+0.18$	0.50	0.34	0.62	-0.018	-0.03	-0.39
Amazon-Ratings $C=5$	1	$+0.07$	0.50	2.00	0.32	-0.270	-0.26	$\mathbf{-5.54}$
	2	$+0.39$	0.24	0.24	0.53	-0.073	-0.13	-0.58
	3	-0.18	0.01	0.01	0.61	$+0.000$	-0.54	-0.48

Table 13: Effect of surrogate activation on embedding quality (ogbn-arxiv, surrogate and suspect both GCN, surrogate depth $K=1$, suspect depth $N=2$). θ: achieved cosine alignment ($\approx$1.0: succeeded; <0.5: failed).

Surrogate Activation	$\theta\uparrow$	Benign $\log_{10} p\uparrow$	WM $\log_{10} p\downarrow$
None (linear)	$\approx$1.0	-0.29	$\mathbf{-66}$
ReLU	<0.5	-0.15	-0.32
Leaky ReLU	<0.5	-0.55	-0.67

(iii) Surrogate embedding dimension d_s. The surrogate's embedding dimension d_s from §3.2 sets the carrier dimension $u_c \in \mathbb{R}^{d_s}$, with d the input feature dimension. When $d_s > d$, $\Delta\mathbf{H}_v$ is rank-bounded by d, so its cosine with a generic u_c is capped at $\sqrt{d/d_s}$ and cannot reach the target θ. We therefore set $d_s = d$; for ogbn-arxiv this is $d_s = 128$. Table 14 sweeps $d_s \in \{32, 64, 128, 256\}$ to verify this: at $d_s = 256 > d$, the achieved alignment $\theta_{\text{ach}} = 0.708 \approx \sqrt{128/256}$ confirms the cap, and the WM detection score collapses from $\log_{10} p = -34.60$ at $d_s = d$ to -0.87, above the rejection threshold $\kappa = -1.30$.

F Black-box Detection via Distillation

Following the black-box decision rule of §3.3, this appendix specifies a two-stage protocol to distill the student $\tilde{f}$ from a suspect f_θ.

Distilling a student $\tilde{f}$ requires committing to a single training dataset, but the suspect's actual training data is unknown to the owner: it may be the benign dataset or one of the B buyers' watermarked datasets, and in the latter case the buyer identity $i^\star$ is also unknown. Stage 1 identifies the most likely training dataset among the $B+1$ candidates; Stage 2 then distills $\tilde{f}$ on it for the white-box pipeline of §3.3. Both stages use the owner's watermarked nodes V_w to probe the f_θ.

Stage 1: Distillation dataset selection. According to membership attack methodology, a trained model's loss on a data point is a strong signal of whether that point was in the training set. Here, f_θ trained on watermarked datasets should fit V_w better than its benign nodes, giving a positive loss-gap $\delta_v = \ell(f_\theta(\mathbf{x}_{\text{orig}})_v, y_v) - \ell(f_\theta(\mathbf{x}_{\text{wm}})_v, y_v)$. Empirical performance is reported in Table 15.

For each buyer candidate i, we score the suspect by its mean loss-gap $\overline{\Delta\ell}_i = \frac{1}{|V_w^{(i)}|}\sum_{v\in V_w^{(i)}} \delta_v^{(i)}$ on $V_w^{(i)}$, and select $\hat{i} = \arg\max_i \overline{\Delta\ell}_i$. A watermarked (WM) suspect's own buyer dominates with $\overline{\Delta\ell}_{i^\star} \gg 0$; a benign suspect yields noise-level scores approximately zero. To decide whether Stage 2 should distill on a watermarked dataset $\mathbf{D}_{\text{wm}}^{(\hat{i})}$ or on the benign dataset, we threshold the peak score $\max_i \overline{\Delta\ell}_i$ by the $\tau_d = 0.5$: $\max_i \overline{\Delta\ell}_i > \tau_d$ commits Stage 2 to $\mathbf{D}_{\text{wm}}^{(\hat{i})}$, otherwise to $\mathbf{D}$. Stage 1 provides empirical evidence the watermark signal is recoverable from f_θ's outputs.

Stage 2: Distillation. We distill a student GCN $\tilde{f}$ from f_θ using two queries, $f_\theta(\mathbf{x}_{\text{orig}})$ and $f_\theta(\mathbf{x}_{\text{wm}})$. Let $\delta_v(g) = g(\mathbf{x}_{\text{wm}})_v - g(\mathbf{x}_{\text{orig}})_v$ denote the logit shift of model g at node v. The objective matches the suspect's logit shift on V_w, anchored by a cross-entropy loss on $\mathbf{x}_{\text{orig}}$:

$$\mathcal{L} = \alpha \cdot \frac{1}{|V_w|} \sum_{v\in V_w} \left\|\delta_v(\tilde{f}) - \delta_v(f_\theta)\right\|_2^2 + (1-\alpha)\,\mathcal{L}_{\text{CE}}\Big(\tilde{f}(\mathbf{x}_{\text{orig}}),\, \mathbf{y}\Big). \tag{53}$$

Table 14: Effect of surrogate embedding dimension d_s on ogbn-arxiv (2-layer GCN, $\delta = 0.5$, set stop alignment target $\theta = 0.96$).

		Detectability $\log_{10}(p)$		**Harmfulness**		**Stealthiness**		
d_s	$\theta\uparrow$	Benign $\uparrow$	WM $\downarrow$	$H\downarrow$	$\hat{H}\downarrow$	$\mathcal{S}\downarrow$	$\bar{\mathcal{S}}\downarrow$	$\mathcal{S}_{\text{knn}}\downarrow$
32	0.964	-0.24	-3.25	0.57	-0.01	0.50	0.52	1.12
64	0.960	-0.35	-20.70	0.42	-0.14	0.50	1.63	1.85
128	0.960	-0.16	-34.60	0.38	-0.17	0.50	1.55	1.81
256	0.708	-0.63	-0.87	0.53	-0.01	0.50	0.29	1.05

Table 15: Distillation dataset selection on 50 WM + 50 benign suspects per dataset (2-layer GCN). Per-suspect peak score $\max_i \overline{\Delta\ell}_i$ (mean $\pm$ std). A_{sel} is distillation-dataset selection accuracy.

Dataset	**Suspect**	$\overline{\Delta\ell}$ on V_w	$\boldsymbol{A_{\text{sel}}}$
ogbn-arxiv	WM ($n=50$)	$\mathbf{+1.003 \pm 0.031}$	$\mathbf{50/50}$
	Benign ($n=50$)	$+0.010 \pm 0.004$	$\mathbf{50/50}$
Amazon-Ratings	WM ($n=50$)	$\mathbf{+1.326 \pm 0.091}$	$\mathbf{50/50}$
	Benign ($n=50$)	$+0.117 \pm 0.028$	$\mathbf{50/50}$

Distilling on the logit-shift residual δ_v subtracts the carrier correlation already present in $\mathbf{x}_{\text{wm}}$, so $\tilde{f}$ learns only the *additional* deviation response signal, which leaves $\tilde{f}$ free of systematic carrier alignment under H_0, so the white-box cosine test of §3.3 on $\tilde{f}$'s weights still inherits the null of Proposition 1.

G Single-Buyer Results ($B = 1$): Watermark Detection

All tables/figures here use the single-buyer decision rule ($\kappa = \log_{10}(0.05) = -1.30$) and the same watermark-embedding fidelity budget $|\Delta\text{Acc}| \leq 1\%$ on benign test nodes. This section covers detection vs. baselines on all datasets and cross-architecture suspects (§G.1), and watermark-injection hyperparameter ablations (§G.2).

G.1 Per-Key Performance vs Baselines

Detection and Fidelity Budget. Table 16 reports detection across all five benchmarks under the fidelity budget $|\Delta\text{Acc}| \leq 1\%$. **Ours** is the only method reaching $\text{AUC} = 1.00$ and $\text{TPR}/\text{FPR} = 10/10\,|\,0/10$ on every dataset, with matching white-box and black-box variants and low harmfulness and stealthiness, whereas the baselines that undercut it on either metric fail to detect. The trigger baselines are miscalibrated: *ImpWM* saturates false positives at $10/10$ with near-random AUC on every dataset, *Backdoor* false-positives up to $10/10$ on ogbn-products, and *BkdOwn* false-positives at $5/10$ on Amazon-Ratings and exhausts memory on the two large graphs despite injecting thousands of nodes and edges, where its deliberately designed trigger features sit at the dense feature-space centroid and therefore score a low $\mathcal{S}_{\text{knn}}$. The key baselines fail differently: *RandPerturb* leaves too weak a signal, with TPR down to $0/10$, and *Data Taggants* degrades at scale, dropping to $4/10$ on ogbn-products and $6/10$ on Reddit. Only Ours delivers reliable and well-calibrated detection within budget at every scale.

Cross-architecture. To assess detection under architectural mismatch, we fix the watermark on a 1-layer GCN surrogate and vary the suspect to 3-layer GAT and GraphSAGE on ogbn-arxiv (Table 17). **Ours** transfers reliably to both, reaching $\text{AUC} = 1.00$ and $\text{TPR}/\text{FPR} = 10/10\,|\,0/10$ with $\Delta N = \Delta E = 0$, low harm, and the lowest stealthiness $\bar{\mathcal{S}} = 1.52$, and its white-box and black-box variants match. The baselines either fail or pay a cost: *RandPerturb* leaves too weak a signal, its TPR only $1/10$ on GAT and $0/10$ on GraphSAGE, *ImpWM* saturates false positives at $10/10$ with near-random AUC on both, *BkdOwn* detects only by injecting 32K nodes and 56K edges at maximal harm $H = 1.00$, *Backdoor* detects with higher harm and variance, and *Data Taggants* transfers but is far less stealthy, $\bar{\mathcal{S}} = 5.33$ against Ours' 1.52. Across eight architectures with deliberately shifted depths, GCN, GAT, Graph Transformer, GraphSAGE, GIN, 8-layer GCNII/ResGCN, and APPNP with personalized PageRank propagation, **Ours** remains reliably detectable even at marketplace scale (§H.1, Table 22).

Table 16: Single-buyer watermark detection on 2-layer **GCN**. **Detectability** (f^w/f^b on watermarked/benign suspect): $\log_{10}(p)$ for Ours/RandPerturb/Data Taggants (↓, fire if $< \kappa = \log_{10}(0.05)$), VSR(%) for Backdoor/ImpWM/BkdOwn (↑, fire if $\kappa > 1/C$). On ogbn-products and Reddit, BkdOwn's per-node 2-hop subgraph cache scales as $O(N \cdot \bar{k}_2 \cdot F)$ and does not fit GPU memory at this scale, leaving its detection cells "—". **Bold** = best. mean$_{\text{std}}$ cell: the mean value with standard deviation over the $n = 10$ seeds.

		Fidelity budget	Detectability			Harmfulness		Stealthiness					
Dataset	**Method**	$\lvert\Delta\text{Acc}\rvert \le 1\%$	f^w/f^b	AUC↑	TPR↑/FPR↓ @ κ	H↓	$\hat{H}$↓	$\mathcal{S}$↓	$\bar{\mathcal{S}}$↓	$\mathcal{S}_{\text{knn}}$↓	$\lvert V_w\rvert$↓	ΔN↓	ΔE↓
Amazon-Ratings	RandPerturb	$-0.51_{0.10}$	$-0.93_{0.67}$ / $-0.63_{0.28}$	0.55	5/10 | **0/10**	$0.66_{0.06}$	$+0.07_{0.06}$	$\mathbf{0.20_{0.00}}$	$2.00_{0.00}$	$2.03_{0.00}$	500	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.39_{0.25}$	$54.14_{7.85}$ / $22.92_{18.13}$	0.89	**10/10** | 3/10	$0.72_{0.05}$	$+0.24_{0.05}$	$0.86_{0.18}$	$1.82_{1.12}$	$1.87_{1.05}$	500	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$+0.04_{0.38}$	$99.99_{0.03}$ / $100.00_{0.00}$	0.45	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$1.39_{0.17}$	**5**	1148	4592
	BkdOwn (Li et al., 2023)	$-0.51_{0.42}$	$98.41_{1.15}$ / $21.62_{4.91}$	**1.00**	**10/10** | 5/10	$0.99_{0.01}$	$+0.99_{0.01}$	—	—	$\mathbf{0.41_{0.02}}$	500	2000	3500
	Data Taggants (Bouaziz et al., 2025)	$-0.72_{0.68}$	$-4.40_{1.48}$ / $-0.36_{0.34}$	**1.00**	**10/10** | **0/10**	$0.32_{0.05}$	$\mathbf{-0.29_{0.08}}$	$0.50_{0.00}$	$7.66_{0.27}$	$7.51_{0.27}$	200	**0**	**0**
	Ours-WB	$-0.64_{0.54}$	$-3.67_{1.18}$ / $-0.42_{0.35}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.31_{0.05}}$	$-0.28_{0.06}$	$0.50_{0.00}$	$\mathbf{1.63_{0.23}}$	$1.70_{0.22}$	500	**0**	**0**
	Ours-BB	$-0.64_{0.54}$	$-3.00_{1.44}$ / $-0.55_{0.40}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.31_{0.05}}$	$-0.28_{0.06}$	$0.50_{0.00}$	$\mathbf{1.63_{0.23}}$	$1.70_{0.22}$	500	**0**	**0**
ogbn-arxiv	RandPerturb	$-0.19_{0.30}$	$-0.34_{0.17}$ / $-0.29_{0.22}$	0.60	0/10 | **0/10**	$0.70_{0.01}$	$+0.05_{0.01}$	$\mathbf{0.50_{0.00}}$	$3.26_{0.00}$	$3.25_{0.00}$	3929	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.06_{0.29}$	$31.13_{4.77}$ / $0.07_{0.08}$	**1.00**	**10/10** | **0/10**	$0.57_{0.02}$	$+0.16_{0.02}$	$1.32_{0.22}$	$2.06_{0.91}$	$2.14_{0.77}$	3908	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$-0.70_{0.19}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$59.79_{6.84}$	**40**	7938	31752
	BkdOwn (Li et al., 2023)	$-0.61_{0.57}$	$98.46_{0.73}$ / $0.00_{0.00}$	**1.00**	**10/10** | **0/10**	$0.99_{0.01}$	$+0.99_{0.01}$	—	—	$\mathbf{0.99_{0.01}}$	8000	32000	56000
	Data Taggants (Bouaziz et al., 2025)	$-0.97_{0.86}$	$-11.85_{4.31}$ / $-0.13_{0.27}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.16_{0.05}}$	$\mathbf{-0.35_{0.06}}$	$\mathbf{0.50_{0.00}}$	$5.33_{0.08}$	$5.05_{0.07}$	3920	**0**	**0**
	Ours-WB	$-0.19_{0.46}$	$-36.80_{3.70}$ / $-0.43_{0.40}$	**1.00**	**10/10** | **0/10**	$0.44_{0.02}$	$-0.21_{0.02}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3929	**0**	**0**
	Ours-BB	$-0.19_{0.46}$	$-21.14_{3.75}$ / $-0.49_{0.45}$	**1.00**	**10/10** | **0/10**	$0.44_{0.02}$	$-0.21_{0.02}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3929	**0**	**0**
Roman-Empire	RandPerturb	$-0.49_{0.52}$	$-0.63_{0.37}$ / $-0.32_{0.27}$	0.79	0/10 | **0/10**	$0.46_{0.02}$	$+0.05_{0.03}$	$\mathbf{0.15_{0.00}}$	$\mathbf{1.50_{0.00}}$	$1.56_{0.00}$	1800	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.49_{0.41}$	$75.53_{0.60}$ / $6.45_{10.63}$	**1.00**	**10/10** | 2/10	$0.89_{0.05}$	$+0.53_{0.06}$	$1.22_{0.22}$	$1.87_{1.09}$	$1.91_{1.05}$	600	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$-0.16_{0.47}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$1.35_{0.08}$	**18**	1062	4248
	BkdOwn (Li et al., 2023)	$-0.60_{0.52}$	$99.97_{0.09}$ / $2.37_{0.59}$	**1.00**	**10/10** | **0/10**	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$\mathbf{0.62_{0.01}}$	1500	6000	10500
	Data Taggants (Bouaziz et al., 2025)	$-0.90_{0.68}$	$-5.63_{1.23}$ / $-0.28_{0.31}$	**1.00**	**10/10** | **0/10**	$0.36_{0.05}$	$-0.07_{0.06}$	$0.50_{0.00}$	$5.90_{0.30}$	$5.79_{0.30}$	200	**0**	**0**
	Ours-WB	$-0.32_{0.62}$	$-11.21_{3.22}$ / $-0.53_{0.47}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.16_{0.01}}$	$\mathbf{-0.26_{0.02}}$	$0.50_{0.00}$	$1.82_{0.11}$	$1.87_{0.11}$	1800	**0**	**0**
	Ours-BB	$-0.32_{0.62}$	$-13.34_{2.71}$ / $-0.62_{0.53}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.16_{0.01}}$	$\mathbf{-0.26_{0.02}}$	$0.50_{0.00}$	$1.82_{0.11}$	$1.87_{0.11}$	1800	**0**	**0**
ogbn-products	RandPerturb	$-0.10_{0.33}$	$-0.59_{0.39}$ / $-0.48_{0.22}$	0.54	1/10 | **0/10**	$0.47_{0.01}$	$+0.00_{0.01}$	$\mathbf{0.50_{0.00}}$	$2.88_{0.00}$	$5.44_{0.07}$	3649	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.31_{0.34}$	$4.32_{1.17}$ / $3.61_{0.52}$	0.69	**10/10** | 10/10	$\mathbf{0.28_{0.01}}$	$+0.01_{0.01}$	$48.53_{23.66}$	$3.66_{0.68}$	$4.90_{0.54}$	2500	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$-0.02_{0.29}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$10.25_{0.52}$	**47**	5874	23496
	BkdOwn (Li et al., 2023)	—	—	—	—	—	—	—	—	—	2500	10000	17500
	Data Taggants (Bouaziz et al., 2025)	$-0.55_{0.50}$	$-1.23_{1.43}$ / $-0.21_{0.33}$	0.71	4/10 | **0/10**	$0.33_{0.11}$	$\mathbf{-0.02_{0.01}}$	$0.50_{0.00}$	$4.87_{0.08}$	$6.61_{0.21}$	6000	**0**	**0**
	Ours-WB	$-0.71_{0.45}$	$-7.01_{2.58}$ / $-0.37_{0.30}$	**1.00**	**10/10** | **0/10**	$0.38_{0.00}$	$-0.01_{0.01}$	$\mathbf{0.50_{0.00}}$	$\mathbf{0.99_{0.26}}$	$\mathbf{4.44_{0.06}}$	15409	**0**	**0**
	Ours-BB	$-0.71_{0.45}$	$-7.22_{2.10}$ / $-0.43_{0.34}$	**1.00**	**10/10** | **0/10**	$0.38_{0.00}$	$-0.01_{0.01}$	$\mathbf{0.50_{0.00}}$	$\mathbf{0.99_{0.26}}$	$\mathbf{4.44_{0.06}}$	15409	**0**	**0**
Reddit	RandPerturb	$+0.42_{0.79}$	$-0.71_{0.29}$ / $-0.41_{0.23}$	0.82	0/10 | **0/10**	$0.07_{0.00}$	$\mathbf{-0.01_{0.02}}$	$0.50_{0.00}$	$7.08_{0.00}$	$19.12_{0.09}$	4088	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.01_{0.81}$	$2.71_{0.83}$ / $0.10_{0.06}$	**1.00**	6/10 | **0/10**	$0.07_{0.01}$	$+0.02_{0.02}$	$13.48_{4.48}$	$4.28_{0.61}$	$17.71_{0.29}$	1500	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$+0.38_{0.82}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$\mathbf{15.22_{1.23}}$	**41**	2812	11248
	BkdOwn (Li et al., 2023)	—	—	—	—	—	—	—	—	—	1500	6000	10500
	Data Taggants (Bouaziz et al., 2025)	$-0.61_{0.36}$	$-1.80_{1.40}$ / $-0.07_{0.20}$	0.93	6/10 | **0/10**	$\mathbf{0.05_{0.01}}$	$\mathbf{-0.01_{0.01}}$	$0.50_{0.00}$	$9.69_{0.82}$	$20.21_{0.78}$	2000	**0**	**0**
	Ours-WB	$+0.27_{0.87}$	$-10.43_{3.36}$ / $-0.34_{0.22}$	**1.00**	**10/10** | **0/10**	$0.07_{0.00}$	$\mathbf{-0.01_{0.02}}$	$\mathbf{0.49_{0.01}}$	$\mathbf{1.11_{0.08}}$	$17.72_{0.04}$	11640	**0**	**0**
	Ours-BB	$+0.27_{0.87}$	$-15.06_{3.19}$ / $-0.40_{0.25}$	**1.00**	**10/10** | **0/10**	$0.07_{0.00}$	$\mathbf{-0.01_{0.02}}$	$\mathbf{0.49_{0.01}}$	$\mathbf{1.11_{0.08}}$	$17.72_{0.04}$	11640	**0**	**0**

Table 17: Cross-architecture single-buyer watermark detection on **ogbn-arxiv** with 3-layer **GAT** and **GraphSAGE** suspects. **Detectability** (f^w/f^b on watermarked/benign suspect): $\log_{10}(p)$ for Ours/RandPerturb/Data Taggants (↓, fire if $< \kappa = \log_{10}(0.05)$), VSR(%) for Backdoor/ImpWM/BkdOwn (↑, fire if $\kappa > 1/C$). **Bold** = best. mean$_{\text{std}}$ cell: the mean value with standard deviation over the $n = 10$ seeds.

		Fidelity budget	Detectability			Harmfulness		Stealthiness					
Suspect	**Method**	$\lvert\Delta\text{Acc}\rvert \le 1\%$	f^w/f^b	AUC↑	TPR↑/FPR↓ @ κ	H↓	$\hat{H}$↓	$\mathcal{S}$↓	$\bar{\mathcal{S}}$↓	$\mathcal{S}_{\text{knn}}$↓	$\lvert V_w\rvert$↓	ΔN↓	ΔE↓
GAT	RandPerturb	$+0.06_{0.35}$	$-0.79_{0.49}$ / $-0.61_{0.34}$	0.61	1/10 | **0/10**	$0.73_{0.01}$	$+0.07_{0.01}$	$2.00_{0.00}$	$13.06_{0.00}$	$12.65_{0.00}$	3905	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.80_{0.32}$	$36.69_{1.63}$ / $0.08_{0.06}$	**1.00**	**10/10** | **0/10**	$0.61_{0.02}$	$+0.20_{0.02}$	$1.36_{0.22}$	$2.06_{0.91}$	$2.14_{0.77}$	3908	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$-0.03_{0.30}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$10.21_{0.88}$	**40**	3907	15628
	BkdOwn (Li et al., 2023)	$-0.86_{0.68}$	$99.88_{0.29}$ / $0.00_{0.00}$	**1.00**	**10/10** | **0/10**	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$\mathbf{0.99_{0.01}}$	8000	32000	56000
	Data Taggants (Bouaziz et al., 2025)	$-0.26_{0.38}$	$-5.59_{2.67}$ / $-0.26_{0.32}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.31_{0.08}}$	$-0.20_{0.03}$	$0.50_{0.00}$	$5.33_{0.07}$	$5.05_{0.07}$	3929	**0**	**0**
	Ours-WB	$-0.16_{0.37}$	$-31.63_{4.63}$ / $-0.45_{0.37}$	**1.00**	**10/10** | **0/10**	$0.43_{0.01}$	$\mathbf{-0.22_{0.01}}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3905	**0**	**0**
	Ours-BB	$-0.16_{0.37}$	$-16.28_{2.38}$ / $-0.59_{0.42}$	**1.00**	**10/10** | **0/10**	$0.43_{0.01}$	$\mathbf{-0.22_{0.01}}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3905	**0**	**0**
Graph-SAGE	RandPerturb	$-0.10_{0.38}$	$-0.67_{0.33}$ / $-0.36_{0.22}$	0.78	0/10 | **0/10**	$0.72_{0.02}$	$+0.05_{0.01}$	$\mathbf{0.50_{0.00}}$	$3.26_{0.00}$	$3.25_{0.00}$	3905	**0**	**0**
	Backdoor (Xu et al., 2023)	$-0.10_{0.61}$	$75.70_{33.46}$ / $0.02_{0.06}$	**1.00**	**10/10** | **0/10**	$0.86_{0.18}$	$+0.45_{0.18}$	$1.29_{0.22}$	$2.06_{0.91}$	$2.14_{0.77}$	3908	**0**	**0**
	ImpWM (Zhang et al., 2024a)	$+0.21_{1.25}$	$100.00_{0.00}$ / $100.00_{0.00}$	0.50	**10/10** | 10/10	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$17.01_{1.25}$	**40**	3907	15628
	BkdOwn (Li et al., 2023)	$-0.56_{0.68}$	$99.63_{0.25}$ / $0.00_{0.00}$	**1.00**	**10/10** | **0/10**	$1.00_{0.00}$	$+1.00_{0.00}$	—	—	$\mathbf{0.99_{0.01}}$	8000	32000	56000
	Data Taggants (Bouaziz et al., 2025)	$-0.05_{0.53}$	$-8.66_{3.14}$ / $-0.20_{0.30}$	**1.00**	**10/10** | **0/10**	$\mathbf{0.07_{0.02}}$	$\mathbf{-0.45_{0.11}}$	$0.50_{0.00}$	$5.37_{0.07}$	$5.08_{0.07}$	3000	**0**	**0**
	Ours-WB	$-0.08_{0.44}$	$-15.62_{4.84}$ / $-0.21_{0.18}$	**1.00**	**10/10** | **0/10**	$0.47_{0.09}$	$-0.20_{0.07}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3905	**0**	**0**
	Ours-BB	$-0.08_{0.44}$	$-8.04_{2.49}$ / $-0.28_{0.20}$	**1.00**	**10/10** | **0/10**	$0.47_{0.09}$	$-0.20_{0.07}$	$\mathbf{0.50_{0.00}}$	$\mathbf{1.52_{0.07}}$	$1.80_{0.05}$	3905	**0**	**0**

G.2 Watermark-Injection Hyperparameter Ablations

Watermark node count $|V_w|$. $|V_w|$ sets the size of the watermark node subset; raising $|V_w|$ amplifies μ in Cor. 1 and hence the detection signal. Table 18 sweeps $|V_w| \in \{40, \ldots, 8000\}$ on ogbn-arxiv. Detection strengthens monotonically with $|V_w|$ ($\log_{10} p$ $-2.08 \rightarrow \mathbf{-52.44}$), with fidelity within $\pm 1\%$ throughout.

Alignment target θ. θ is the target cosine $\cos(\Delta\mathbf{H}_v, u_{y_v})$ enforced by Algorithm 1's stop criterion at injection time; raising θ amplifies μ in Cor. 1 and hence the detection signal. Table 19 sweeps $\theta \in \{0.60, \ldots, 0.97\}$ on ogbn-arxiv. Detection strengthens monotonically with θ ($\log_{10} p$ $-10.45 \rightarrow$ $\mathbf{-35.27}$), saturating near $\theta \approx 0.90$, with fidelity within $\pm 1\%$ throughout.

Perturbation bound δ. δ caps the per-node ℓ_∞ feature perturbation; raising δ amplifies μ in Cor. 1, trading against stealth and harmfulness. Table 20 sweeps $\delta \in \{0.01, \ldots, \infty\}$ on ogbn-arxiv. Detection strengthens with δ up to $\delta = 1.0$ ($\log_{10} p$ $-2.98 \rightarrow \mathbf{-48.50}$), with fidelity within $\pm 1\%$; harmfulness and stealthiness degrade beyond $\delta = 0.1$.

Loss regularisation λ_1, λ_2. λ_1 penalises feature perturbation magnitude, and λ_2 constrains the embedding change orthogonal to the secret direction. Table 21 ablates $\lambda_1, \lambda_2 \in \{0, 0.1\}$ on ogbn-arxiv. λ_1 drives detectability and stealthiness at a small harmfulness cost, while λ_2 bounds the

Table 18: Effect of watermark node count $|V_w|$ on ogbn-arxiv (2-layer GCN; $\theta = 0.96$, $\delta = 0.5$ fixed).

	Fidelity	**Detectability** $\log_{10}(p)$		**Harmfulness**		**Stealthiness**	
$\lvert V_w\rvert$	ΔAcc (%)	Benign ↑	WM ↓	H↓	$\hat{H}$↓	$\mathcal{S}$↓	$\bar{\mathcal{S}}$↓
40	−0.11	−1.23	−2.08	0.62	−0.050	0.50	1.42
120	+0.06	−1.23	−1.72	0.58	−0.033	0.50	1.46
200	−0.21	**−0.45**	−2.88	0.58	−0.045	0.50	1.50
320	−0.24	**−0.45**	−3.05	0.60	−0.053	0.50	1.49
400	−0.73	−1.23	−5.56	0.60	−0.055	0.50	1.54
800	−0.46	−1.23	−9.89	0.55	−0.088	0.50	1.66
2000	**+0.27**	**−0.45**	−18.04	0.51	−0.135	0.50	1.61
4000	+0.07	−0.48	−33.29	0.49	−0.175	0.50	1.43
8000	−0.30	−0.46	**−52.44**	**0.382**	**−0.259**	0.50	**1.26**

Table 19: Effect of alignment target θ on ogbn-arxiv (2-layer GCN; $|V_w| = 4000$, $\delta = 0.5$ fixed).

	Fidelity	**Detectability** $\log_{10}(p)$		**Harmfulness**		**Stealthiness**	
θ	ΔAcc (%)	Benign ↑	WM ↓	H↓	$\hat{H}$↓	$\mathcal{S}$↓	$\bar{\mathcal{S}}$↓
0.60	−0.18	**−0.28**	−10.45	0.60	−0.046	0.50	**0.54**
0.70	**+0.30**	−0.72	−18.39	0.57	−0.072	0.50	0.76
0.80	−0.43	−0.35	−30.40	0.42	−0.226	0.50	1.54
0.90	−0.35	−0.60	−34.71	**0.37**	**−0.275**	0.50	2.13
0.95	+0.17	−0.40	−35.16	0.40	−0.239	0.50	1.60
0.97	−0.24	−0.49	**−35.27**	0.43	−0.210	0.50	1.44

orthogonal embedding drift to safeguard predictions, and the deployed $\lambda_1 = \lambda_2 = 0.1$ yields the smallest accuracy drop with harmfulness kept low.

H Marketplace Results ($B > 1$): Owner Attribution

Companion to the main paper's marketplace experiments (§4.2, §4.3, §4.3). All results in this section are evaluated at $B = 1000$ buyers and report Owner Attribution Accuracy (OAA) as the primary metric, under the watermark-embedding fidelity budget $|\Delta\text{Acc}| \leq 1\%$ on benign test nodes. We cover cross-architecture OAA analysis (§H.1), the fidelity budget vs. OAA trade-off (§H.2), attack utility cost under removal attacks (§H.3), scalability of owner attribution across buyer pool size B (§H.4), cross-probe owner attribution (§H.5), and runtime and computational cost (§H.6).

H.1 Cross-Architecture Multi-Buyer Attribution

Tables 22 and 23 extend Table 1's $B = 1000$ attribution test from the default 2-layer GCN suspect to seven additional architectures: GAT-2L, Graph Transformer-2L, GraphSAGE-2L, GIN-2L, GCNII-8L, ResGCN-8L, and APPNP-2L with 10 propagation steps. Table 22 uses the homophilic ogbn-arxiv benchmark and Table 23 the heterophilic Roman-Empire benchmark. We report Ours-WB, Ours-BB, *Backdoor*, and *Data Taggants*. *RandPerturb* is excluded because it yields chance-level WDA and OAA on every dataset and mode, and *BkdOwn* and *ImpWM* remain excluded as in §4.2. The GCN-2L block in each table reuses the corresponding numbers from Table 1 as the surrogate-matched reference.

Ours transfers uniformly across architectures. *Backdoor* flags a suspect as watermarked when the largest VSR over the probe pool exceeds the $1/C$ chance threshold. With $B = 1000$ probes a benign model almost always produces at least one probe above $1/C$ by chance, so benign suspects are misclassified as watermarked and *Backdoor*'s WDA falls to the 50% chance level on every architecture. Its trigger injection also inflates harm to $H \approx 1.0$. Ours instead holds $H, \hat{H}, S$ low at the GCN-reference level across all suspects and independent of the suspect architecture, because the watermark lives in feature space rather than in any architecture-specific component. On every architecture at least one of Ours-WB and Ours-BB attains $\geq 89\%$ WDA on ogbn-arxiv and $\geq 95\%$ WDA on Roman-Empire, in both cases with $\geq 90\%$ OAA. *Data Taggants* transfers to most architectures but, under the 1% fidelity budget, its attribution is architecture-dependent, reaching 90% OAA on several suspects yet collapsing to as low as 2% on GIN and 8% on GCNII.

Table 20: Effect of per-node perturbation bound δ on ogbn-arxiv (2-layer GCN; $|V_w|=8000$, $\theta=0.96$ fixed). ∞: no clipping constraint.

	Fidelity	**Detectability** $\log_{10}(p)$		**Harmfulness**		**Stealthiness**		
δ	ΔAcc (%)	Benign ↑	WM ↓	$H\downarrow$	$\hat{H}\downarrow$	$\mathcal{S}\downarrow$	$\bar{\mathcal{S}}\downarrow$	$\mathcal{S}_{\text{knn}}\downarrow$
0.01	-0.49	$\mathbf{-0.13}$	-2.98	0.66	$+0.010$	**0.01**	**0.09**	**1.02**
0.1	-0.80	-0.21	-24.13	0.46	-0.190	0.10	0.67	1.21
0.5	-0.26	-0.14	-46.20	0.25	-0.397	0.50	3.44	3.45
1.0	$\mathbf{+0.60}$	-0.34	$\mathbf{-48.50}$	0.24	-0.408	1.00	5.35	5.27
3.0	$+0.37$	-0.20	-40.15	0.24	-0.413	3.00	6.32	6.21
5.0	$+0.22$	-0.22	-39.56	**0.23**	$\mathbf{-0.423}$	4.98	6.33	6.22
∞	$+0.19$	-0.16	-42.63	**0.23**	-0.416	5.02	6.36	6.25

Table 21: Ablation of loss regularization terms λ_1 and λ_2 (Eq. 1) on ogbn-arxiv (2-layer GCN; $|V_w|=8000$, $\theta=0.96$ fixed). $\lambda_1, \lambda_2 \in \{0, 0.1\}$; 0 = disabled.

		Fidelity	**Detectability** $\log_{10}(p)$		**Harmfulness**		**Stealthiness**		
λ_1	λ_2	ΔAcc (%)	Benign ↑	WM ↓	$H\downarrow$	$\hat{H}\downarrow$	$\mathcal{S}\downarrow$	$\bar{\mathcal{S}}\downarrow$	$\mathcal{S}_{\text{knn}}\downarrow$
0	0	-0.64	-0.11	-37.98	**0.20**	$\mathbf{-0.34}$	0.50	3.34	3.37
0.1	0	-0.42	-0.43	$\mathbf{-43.81}$	0.32	-0.22	0.50	1.34	1.65
0	0.1	-0.52	-0.11	-37.89	0.21	-0.33	0.50	3.35	3.37
0.1	0.1	$\mathbf{-0.34}$	$\mathbf{-0.11}$	-42.80	0.32	-0.23	0.50	**1.34**	**1.65**

The two benchmarks isolate two routes by which message passing lowers white-box recoverability. The watermark is an embedding shift in input feature space, and Ours-WB recovers it by linearly aligning the suspect's native representation to the surrogate. The low WB cells are exactly those where this alignment loses fidelity, and the homophilic and heterophilic benchmarks isolate two separate routes. On ogbn-arxiv the only weak WB cell is ResGCN-8L, which isolates the depth route, as its plain-residual stack over-smooths the watermark across eight layers while GCNII and APPNP, which re-inject the input at every step, keep WB at or above 94%. On Roman-Empire the shallow aggregation-heavy GraphSAGE, GIN, and Graph Transformer drop as well, which isolates the heterophily route, as their neighbor aggregation now mixes dissimilar-class neighbors and distorts the native embedding geometry that the homophilic graph leaves intact. ResGCN-8L stays weak on both, since depth degrades the watermark signal regardless of homophily.

The watermark itself is never removed, and black-box distillation recovers it. In every weak WB case Ours-BB restores detection by distilling the suspect into a clean 2-layer GCN student, whose shallow representation carries no internal nonlinearity and aligns cleanly to the surrogate, reaching at least 89% WDA with $\geq 90\%$ OAA and $\geq 95\%$ on Roman-Empire. This recovery shows that the watermark survives in the suspect's decision function in all cases, so the weak WB cells reflect native-weight recoverability rather than watermark loss. The one reverse case is arxiv GCNII-8L, where WB reaches $94/98$ but distilling the deep stack costs BB some signal at $86/76$. The two modes are therefore complementary, and their union keeps detection reliable across all eight architectures and both homophily regimes.

H.2 Fidelity Budget vs. Owner Attribution Accuracy

Our studies find that the watermark node count $|V_w|$ is a key knob controlling the trade-off between owner attribution accuracy and fidelity budget. The Amazon-Ratings shortfall in the main marketplace results (Table 1: Ours-WB / Ours-BB $66/30$ and $77/56$) reflects the $|\Delta\text{Acc}| \leq 1\%$ fidelity budget binding at $|V_w|=625$ (watermark rate $\approx 2.6\%$). Figure 7 shows raising the watermark node count to $|V_w|=5000$ (watermark rate $\approx 20.4\%$) improves owner attribution accuracy to $98/96\%$ for Ours-WB / Ours-BB but degrades the fidelity budget to $\approx -7.4\%$.

H.3 Robustness Analysis under Removal Attacks

Attack setup and results on the homophilic graph. We test whether the watermark survives four removal attacks at marketplace scale ($B = 1000$ buyers) across all five datasets. To maintain signal strength under attack, we increase the number of watermark nodes while keeping the fidelity

Table 22: Cross-architecture multi-buyer attribution on **ogbn-arxiv** (homophilic) at $B = 1000$ (50 wm and 50 benign suspects per cell) under fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$. The GCN-2L block reuses the arxiv numbers from Table 1 for direct architecture-shift comparison. Column conventions follow Table 1. Mean$_{\pm\mathrm{std}}$ over 50 wm suspects.

		Detection & Attribution		Harmfulness		Stealthiness		
Architecture	**Method**	WDA (%, ↑)	OAA (%, ↑)	H (↓)	$\hat{H}$ (↓)	$\mathcal{S}$ (↓)	$\bar{\mathcal{S}}$ (↓)	$\mathcal{S}_{\mathrm{knn}}$ (↓)
GCN-2L (ref)	Backdoor (Xu et al., 2023)	50.0	18.0	$0.68_{\pm 0.05}$	$+0.37_{\pm 0.04}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	95.0	90.0	$\mathbf{0.17_{\pm 0.05}}$	$\mathbf{-0.36_{\pm 0.06}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**100.0**	**100.0**	$0.38_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.55_{\pm 0.03}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	94.0	**100.0**	$0.38_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.55_{\pm 0.03}}$	$\mathbf{1.81_{\pm 0.03}}$
GAT-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$0.83_{\pm 0.04}$	$+0.53_{\pm 0.03}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	85.0	70.0	$\mathbf{0.27_{\pm 0.06}}$	$\mathbf{-0.26_{\pm 0.04}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**100.0**	**100.0**	$0.33_{\pm 0.01}$	$-0.21_{\pm 0.02}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	96.0	**100.0**	$0.33_{\pm 0.01}$	$-0.21_{\pm 0.02}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
Graph Transformer-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$1.00_{\pm 0.00}$	$+0.68_{\pm 0.01}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	94.0	88.0	$\mathbf{0.05_{\pm 0.02}}$	$\mathbf{-0.51_{\pm 0.10}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**99.0**	**100.0**	$0.42_{\pm 0.02}$	$-0.15_{\pm 0.02}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	99.0	**100.0**	$0.42_{\pm 0.02}$	$-0.15_{\pm 0.02}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
GraphSAGE-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$1.00_{\pm 0.00}$	$+0.33_{\pm 0.04}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	95.0	90.0	$\mathbf{0.05_{\pm 0.01}}$	$\mathbf{-0.49_{\pm 0.09}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**99.0**	**100.0**	$0.32_{\pm 0.03}$	$-0.22_{\pm 0.04}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	98.0	**100.0**	$0.32_{\pm 0.03}$	$-0.22_{\pm 0.04}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
GIN-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$0.76_{\pm 0.03}$	$+0.35_{\pm 0.03}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	67.0	32.0	$0.53_{\pm 0.30}$	$+0.03_{\pm 0.30}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**95.0**	94.0	$\mathbf{0.34_{\pm 0.01}}$	$\mathbf{-0.16_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.36_{\pm 0.03}}$	$\mathbf{1.66_{\pm 0.02}}$
	Ours-BB	86.0	**100.0**	$\mathbf{0.34_{\pm 0.01}}$	$\mathbf{-0.16_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.36_{\pm 0.03}}$	$\mathbf{1.66_{\pm 0.02}}$
GCNII-8L	Backdoor (Xu et al., 2023)	50.0	20.0	$1.00_{\pm 0.00}$	$+0.64_{\pm 0.00}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	71.0	42.0	$\mathbf{0.29_{\pm 0.08}}$	$\mathbf{-0.30_{\pm 0.06}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**94.0**	**98.0**	$0.56_{\pm 0.01}$	$-0.04_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	86.0	76.0	$0.56_{\pm 0.01}$	$-0.04_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
ResGCN-8L	Backdoor (Xu et al., 2023)	50.0	10.0	$0.71_{\pm 0.06}$	$+0.35_{\pm 0.07}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	**95.0**	92.0	$\mathbf{0.29_{\pm 0.07}}$	$\mathbf{-0.24_{\pm 0.06}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	80.0	58.0	$0.42_{\pm 0.01}$	$-0.12_{\pm 0.05}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.24_{\pm 0.03}}$	$\mathbf{1.58_{\pm 0.02}}$
	Ours-BB	89.0	**100.0**	$0.42_{\pm 0.01}$	$-0.12_{\pm 0.05}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.24_{\pm 0.03}}$	$\mathbf{1.58_{\pm 0.02}}$
APPNP-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$0.99_{\pm 0.00}$	$+0.65_{\pm 0.00}$	$1.77_{\pm 0.17}$	$3.68_{\pm 0.76}$	$3.58_{\pm 0.70}$
	Data Taggants (Bouaziz et al., 2025)	91.0	82.0	$\mathbf{0.11_{\pm 0.03}}$	$\mathbf{-0.44_{\pm 0.08}}$	$\mathbf{0.50_{\pm 0.00}}$	$5.29_{\pm 0.09}$	$5.03_{\pm 0.09}$
	Ours-WB	**100.0**	**100.0**	$0.39_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$
	Ours-BB	96.0	**100.0**	$0.39_{\pm 0.01}$	$-0.17_{\pm 0.01}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.56_{\pm 0.04}}$	$\mathbf{1.81_{\pm 0.03}}$

Table 23: Cross-architecture multi-buyer attribution on **Roman-Empire** (heterophilic) at $B = 1000$ (50 wm and 50 benign suspects per cell) under fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$. The GCN-2L block reuses the Roman-Empire numbers from Table 1 for direct architecture-shift comparison. Column conventions follow Table 1. Mean$_{\pm\mathrm{std}}$ over 50 wm suspects.

		Detection & Attribution		Harmfulness		Stealthiness		
Architecture	**Method**	WDA (%, ↑)	OAA (%, ↑)	H (↓)	$\hat{H}$ (↓)	$\mathcal{S}$ (↓)	$\bar{\mathcal{S}}$ (↓)	$\mathcal{S}_{\mathrm{knn}}$ (↓)
GCN-2L (ref)	Backdoor (Xu et al., 2023)	50.0	20.0	$0.92_{\pm 0.02}$	$+0.57_{\pm 0.03}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.78}$
	Data Taggants (Bouaziz et al., 2025)	80.0	60.0	$0.33_{\pm 0.05}$	$-0.07_{\pm 0.08}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	**100.0**	**100.0**	$\mathbf{0.16_{\pm 0.01}}$	$\mathbf{-0.25_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$
	Ours-BB	98.0	**100.0**	$\mathbf{0.16_{\pm 0.01}}$	$\mathbf{-0.25_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$
GAT-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$0.94_{\pm 0.01}$	$+0.45_{\pm 0.02}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	75.0	52.0	$0.54_{\pm 0.07}$	$+0.00_{\pm 0.07}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	**99.0**	**100.0**	$\mathbf{0.28_{\pm 0.02}}$	$\mathbf{-0.26_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.66_{\pm 0.10}}$	$\mathbf{1.72_{\pm 0.10}}$
	Ours-BB	**99.0**	**100.0**	$\mathbf{0.28_{\pm 0.02}}$	$\mathbf{-0.26_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.66_{\pm 0.10}}$	$\mathbf{1.72_{\pm 0.10}}$
Graph Transformer-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$1.00_{\pm 0.00}$	$+0.89_{\pm 0.01}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	83.0	66.0	$0.10_{\pm 0.03}$	$-0.04_{\pm 0.05}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	70.0	42.0	$\mathbf{0.01_{\pm 0.00}}$	$\mathbf{-0.12_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.72_{\pm 0.09}}$
	Ours-BB	**95.0**	**100.0**	$\mathbf{0.01_{\pm 0.00}}$	$\mathbf{-0.12_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.72_{\pm 0.09}}$
GraphSAGE-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$1.00_{\pm 0.00}$	$+0.89_{\pm 0.01}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	90.0	80.0	$0.10_{\pm 0.03}$	$-0.03_{\pm 0.05}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	71.0	42.0	$\mathbf{0.01_{\pm 0.00}}$	$\mathbf{-0.12_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.10}}$	$\mathbf{1.73_{\pm 0.09}}$
	Ours-BB	**98.0**	**100.0**	$\mathbf{0.01_{\pm 0.00}}$	$\mathbf{-0.12_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.10}}$	$\mathbf{1.73_{\pm 0.09}}$
GIN-2L	Backdoor (Xu et al., 2023)	50.0	12.0	$0.91_{\pm 0.02}$	$+0.59_{\pm 0.03}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	52.0	2.0	$0.37_{\pm 0.05}$	$-0.02_{\pm 0.07}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	84.0	70.0	$\mathbf{0.19_{\pm 0.01}}$	$\mathbf{-0.19_{\pm 0.03}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.68_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$
	Ours-BB	**98.0**	**100.0**	$\mathbf{0.19_{\pm 0.01}}$	$\mathbf{-0.19_{\pm 0.03}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.68_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$
GCNII-8L	Backdoor (Xu et al., 2023)	50.0	22.0	$1.00_{\pm 0.00}$	$+0.72_{\pm 0.01}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	54.0	8.0	$0.38_{\pm 0.07}$	$-0.03_{\pm 0.10}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	**100.0**	**100.0**	$\mathbf{0.04_{\pm 0.01}}$	$\mathbf{-0.36_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.08}}$
	Ours-BB	99.0	**100.0**	$\mathbf{0.04_{\pm 0.01}}$	$\mathbf{-0.36_{\pm 0.01}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.67_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.08}}$
ResGCN-8L	Backdoor (Xu et al., 2023)	50.0	18.0	$0.81_{\pm 0.03}$	$+0.39_{\pm 0.03}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	77.0	56.0	$0.43_{\pm 0.07}$	$-0.07_{\pm 0.05}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	82.0	64.0	$\mathbf{0.36_{\pm 0.01}}$	$\mathbf{-0.15_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.65_{\pm 0.06}}$	$\mathbf{1.71_{\pm 0.06}}$
	Ours-BB	**95.0**	**90.0**	$\mathbf{0.36_{\pm 0.01}}$	$\mathbf{-0.15_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.65_{\pm 0.06}}$	$\mathbf{1.71_{\pm 0.06}}$
APPNP-2L	Backdoor (Xu et al., 2023)	50.0	20.0	$0.99_{\pm 0.01}$	$+0.67_{\pm 0.01}$	$1.34_{\pm 0.17}$	$3.52_{\pm 0.81}$	$3.52_{\pm 0.79}$
	Data Taggants (Bouaziz et al., 2025)	62.0	24.0	$0.29_{\pm 0.05}$	$-0.12_{\pm 0.09}$	$\mathbf{0.50_{\pm 0.00}}$	$5.73_{\pm 0.27}$	$5.62_{\pm 0.26}$
	Ours-WB	**100.0**	**100.0**	$\mathbf{0.09_{\pm 0.01}}$	$\mathbf{-0.32_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.68_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$
	Ours-BB	99.0	**100.0**	$\mathbf{0.09_{\pm 0.01}}$	$\mathbf{-0.32_{\pm 0.02}}$	$\mathbf{0.50_{\pm 0.00}}$	$\mathbf{1.68_{\pm 0.09}}$	$\mathbf{1.73_{\pm 0.09}}$

budget $|\Delta\mathrm{Acc}| \leq 1\%$, where $\Delta\mathrm{Acc}$ denotes the accuracy drop on benign test nodes. Each attack assumes that the adversary has no access to the owner's watermark nodes. (1) *Random perturbation:* add zero-mean Gaussian noise with standard deviation $\sigma \in \{0.1, 0.5\}$ to the feature vectors of a random fraction $r \in \{0.1, 0.2, 0.5\}$ of nodes in $\mathcal{D}_{\mathrm{wm}}$, then retrain a suspect; (2) *Model pruning:* zero a fraction of the suspect's smallest-magnitude weights across all layers; (3) *Fine-tuning:* continue

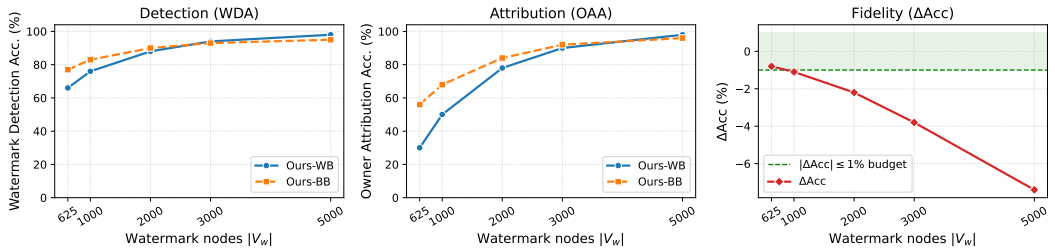


Figure 7: Amazon-Ratings marketplace ($B = 1000$): WDA, OAA, and fidelity (ΔAcc) vs. watermark node count $|V_w|$ for Ours-WB / Ours-BB; larger $|V_w|$ improves attribution but breaches the 1% fidelity budget (green band).

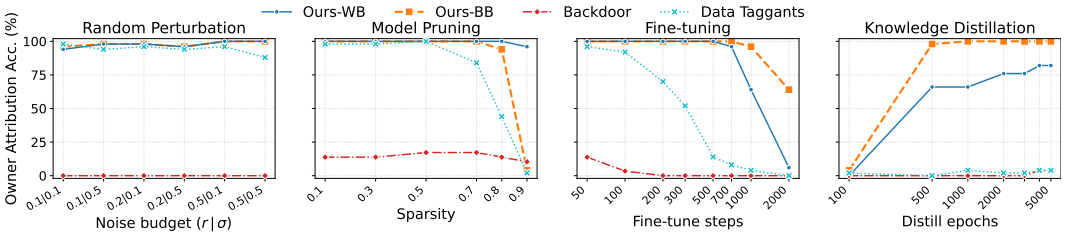


Figure 8: Owner Attribution Accuracy (%) on **ogbn-arxiv** (homophilic) at $B = 1000$ versus four removal attacks. Each curve aggregated over 50 watermarked suspects.

training the suspect on benign nodes for several epochs; (4) *Knowledge distillation:* train a new student model to mimic the suspect's outputs on benign nodes.

Figure 8 reports *Owner Attribution Accuracy (OAA)* on the homophilic dataset ogbn-arxiv. **Ours-WB (white-box setting)** and **Ours-BB (black-box setting)** keep OAA robust across all attacks. Random perturbation barely affects performance, with OAA $\geq 94\%$ across all noise levels, since the attack is blind to watermark nodes. Model pruning is handled by Ours-WB ($\geq 96\%$ up to sparsity 0.9), while Ours-BB collapses only at extreme sparsity (4% at 0.9). Notably, pruning at 0.9 sparsity also causes $\approx 30\%$ utility drop, deterring such extreme pruning (Appendix H.3, Figure 10). Fine-tuning and distillation are handled by Ours-BB (96% at 1000 fine-tune steps, $\geq 98\%$ at ≥ 500 distillation epochs), while Ours-WB degrades. Ours-BB is more robust because it relies on suspect outputs on probe nodes rather than internal parameters, which are more vulnerable to change. Both settings sustain high OAA, falling only at extreme attacker budgets (2000 fine-tune steps: 64%; 100-epoch under-converged distillation: 4%), which are already costly and degrade suspect utility.

In contrast, *Backdoor* never exceeds 17% OAA under any attack, confirming that trigger-based watermarks are vulnerable to removal strategies that disrupt trigger influence. *Data Taggants*, the strongest baseline, resists feature noise (OAA $\geq 88\%$) and light pruning but is erased by distillation (2.6%) and heavy fine-tuning.

Robustness results on heterophilic graphs under Removal Attacks. We extend the above homophilic ogbn-arxiv result to the heterophilic Roman-Empire, at the same $B = 1000$ scale and $\leq 1\%$ fidelity budget, with Ours-WB, Ours-BB, *Backdoor*, and *Data Taggants* methods.

Figure 9 shows that **Ours-WB** and **Ours-BB** remain robust on data-side attacks but weaken on training-side attacks on this heterophilic graph. Random perturbation barely moves either variant and OAA $\geq 94\%$ on the first five (r, σ) levels; the most aggressive level $(r{=}0.5, \sigma{=}0.5)$ drops Ours-BB to 0% but already costs the suspect $\sim 25\%$ accuracy (Figure 11), making it self-destructive for the attacker. Model pruning is handled equally well by Ours-WB and Ours-BB (100% up to sparsity 0.9); pruning at 0.9 sparsity drops the suspect's clean accuracy by approximately 42%. Fine-tuning and knowledge distillation are harder on this heterophilic graph: fine-tuning drops OAA of Ours-WB and Ours-BB at ≥ 300 steps but attack utility cost is already 3% at 300 steps. Knowledge distillation reduces Ours-WB to 8.6% and Ours-BB to 22.3% averaged across epochs although the attack utility cost is high, which indicates that the weaker watermark signal on this smaller heterophilic graph leaves less buffer for the watermark to survive aggressive training-based attacks. In contrast, *Backdoor* never exceeds 10% OAA on any attack (overall 4.1%), confirming that the trigger-cross-fire failure mode of §4.2 carries over directly to the heterophilic regime: attacks only deepen an already-broken

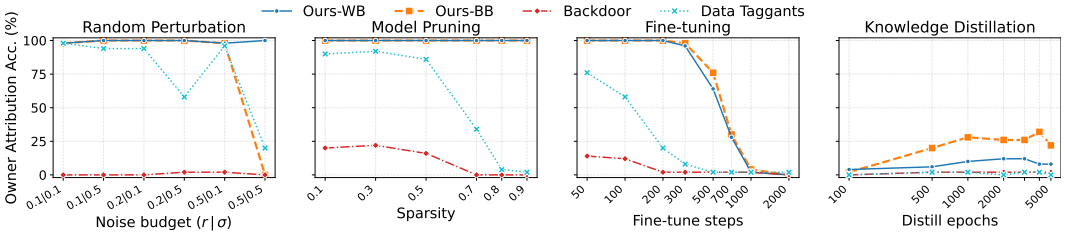


Figure 9: Owner Attribution Accuracy (%) on **Roman-Empire** (heterophilic) at $B = 1000$ versus four removal attacks. Each curve aggregated over 50 watermarked suspects.

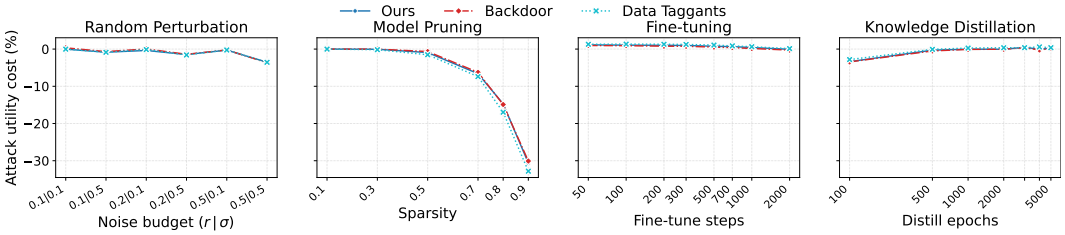


Figure 10: Attack utility cost (%) on **ogbn-arxiv** (homophilic) at $B = 1000$ versus four removal attacks. Each curve aggregated over 50 watermarked suspects.

attribution baseline. *Data Taggants*, the strongest baseline, again resists feature noise and light pruning but is erased by distillation and fine-tuning, reaching 35% OAA overall against *Backdoor*'s 4.1%.

Attack utility cost on the suspect under Removal Attacks. Attack also can degrade the downstream suspect's utility while removing attack can degrade the watermark detection accuracy and owner attribution accuracy. We therefore measure the attack utility cost to quantify the attack's own cost on the suspect's accuracy on the original test nodes, isolating it from the watermark's inherent cost. This is important for understanding the practical implications of the attack: a successful removal attack that significantly degrades the suspect's performance may not be desirable for an attacker, even if it evades detection.

Therefore, we define the $\Delta\mathrm{Acc}(\epsilon) = \mathrm{Acc}(f_\epsilon;\, x_{\mathrm{orig}}) - \mathrm{Acc}(f_\theta;\, x_{\mathrm{orig}})$ as the attack utility cost, where f_ϵ^w is the post-attack suspect, f_θ is the pre-attack suspect model, $\mathrm{Acc}(f_\epsilon;\, x_{\mathrm{orig}})$ denotes the accuracy of the model on the original test nodes. This measures the drop in suspect accuracy after a removal attack, relative to the pre-attack suspect model. Figure 10 and Figure 11 show a high attack utility cost per watermark method as the larger the attack budget, which indicates that the attack is not only removing the watermark but also significantly degrading the suspect's performance on the original task. This highlights the practical trade-off for an attacker: while they may successfully evade detection, they may also be harming the utility of the stolen model, which could be undesirable from their perspective.

Watermark node inspection. Additionally, from the owner's perspective, the watermark nodes in a released watermarked dataset should not be easily identifiable by the attacker; otherwise the attacker could simply remove them and evade detection. We therefore assess how easily an attacker can localize the watermark nodes that each method injects into the released graph, whether they carry a feature perturbation or a trigger pattern.

Following Kerckhoffs's principle (Katz & Lindell, 2014), we grant the attacker full knowledge of the watermarking scheme but withhold the secret key $\mathcal{K}$, and ask whether the released graph alone lets it localize the watermark nodes. We consider two families of statistical inspection on the worse-case scenario that the attacker knows the number $|V_w|$ of watermark nodes and finds the top-$|V_w|$.

(a) **Repeated pattern detection** scans all N nodes of the released graph and flags those whose feature values or whole vectors recur unusually often: *repeated feature values* scores a node v by how many of the N nodes share each of v's feature values, summed across dimensions, and *repeated feature vectors* scores v by how many of the N nodes are exact duplicates of v's entire feature vector. (b) **Abnormal distribution detection** uses five statistics to measure how far each node v deviates

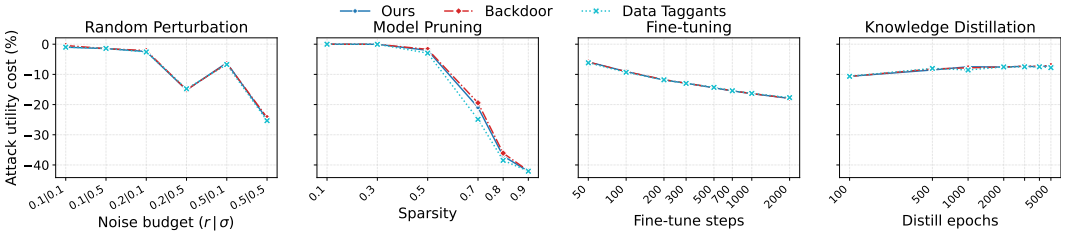


Figure 11: Attack utility cost (%) on **Roman-Empire** (heterophilic) at $B = 1000$ versus four removal attacks. Each curve aggregated over 50 watermarked suspects.

Table 24: Watermark node identification (top-$|V_w|$ inspection). $|V_w| = 7{,}756$ on ogbn-arxiv and $|V_w| = 500$ on Amazon-Ratings. With x the number of watermark nodes among the top-$|V_w|$ nodes flagged by each detector, we report **Correctly identified** (Cor., %) $= x/|V_w|$; lower Cor. is stealthier. **Bold** marks the worst case (highest Cor.) within each method's row. Random selection $= |V_w|/N$ means that the Cor. rate is a random guessing.

		(a) Repeated pattern detection		(b) Abnormal distribution detection				
Dataset	**Method**	Feature values	Feature vectors	Feature magnitude	Neighbor deviation	Feature-space isolation	Manifold deviation	Class-centroid deviation
ogbn-arxiv	RandPerturb	0.0	1.9	**100.0**	87.1	**100.0**	**100.0**	**100.0**
	Backdoor (Xu et al., 2023)	**100**	3.8	99.4	83.4	6.2	3.5	99.7
	ImpWM (Zhang et al., 2024a)	**100.0**	**100.0**	**100.0**	90.1	0.0	12.5	90.0
	BkdOwn (Li et al., 2023)	**100**	**100**	33.4	0.0	3.3	0.2	74.6
	Data Taggants (Bouaziz et al., 2025)	0.0	1.8	95.8	86.0	85.9	97.0	**99.0**
	Ours	0.3	4.2	23.9	25.0	23.5	**40.5**	22.6
	Random selection	4.7	4.7	4.7	4.7	4.7	4.7	4.7
Amazon-Ratings	RandPerturb	0.0	0.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	Backdoor (Xu et al., 2023)	**100**	0.0	98.8	96.4	7.4	2.2	98.8
	ImpWM (Zhang et al., 2024a)	**100.0**	**100.0**	80.1	63.9	0.0	0.0	80.1
	BkdOwn (Li et al., 2023)	**100**	**100**	33.4	0.0	37.2	0.6	33.4
	Data Taggants (Bouaziz et al., 2025)	0.0	0.0	**99.0**	**99.0**	95.5	94.0	**99.0**
	Ours	0.0	0.0	40.6	**53.8**	17.4	46.0	41.4
	Random selection	2.0	2.0	2.0	2.0	2.0	2.0	2.0

from the N released nodes and flags the highest-scoring nodes: *abnormal feature magnitude* is the ℓ_2 norm of v's feature vector; *abnormal neighbor deviation* is v's Euclidean distance, which measures how far v's feature vector deviates from the mean feature vector of v's one-hop graph neighbors; *abnormal feature-space isolation* is v's mean Euclidean distance, which measures how far v's feature vector deviates from v's k nearest neighbors among the N nodes; *abnormal manifold deviation* fits PCA to the N nodes' feature vectors to rebuild each node's full feature vector using only a few principal components, and scores v by the Euclidean distance between v's original and rebuilt feature vectors, which measures how far v's feature vector deviates from the common low-dimensional surface the N nodes share; and *abnormal class-centroid deviation* is v's Euclidean distance, which measures how far v's feature vector deviates from the mean feature vector over the nodes of v's class.

Table 24 summarizes these inspections on the watermarked datasets. The trigger-based methods *Backdoor*, *BkdOwn*, and *ImpWM* expose their watermark nodes by simple counting, as their identical trigger patterns are flagged at 100% by the repeated-value or repeated-vector inspection. The feature-perturbation baselines *RandPerturb* and *Data Taggants* evade the repeated-pattern inspection because their per-node perturbations are continuous and non-repeating, yet the abnormal-distribution inspections still flag 95–100% of their watermark nodes, since the perturbations raise feature magnitude and push the nodes off the class manifold. *Data Taggants* is most exposed by class-centroid deviation at 99%, a direct signature of gradient matching toward off-class random keys. In contrast, Ours evades every inspection angle: the strongest attacker flags at most 53.8% of watermark nodes on Amazon-Ratings and 40.5% on ogbn-arxiv, even with worst-case knowledge of $|V_w|$. Because the correctly identified rate stays far below 100%, the attacker must flag a large fraction of benign nodes to catch the watermark, so it cannot be removed without discarding benign data. This stealth follows from the stealthiness regularizer $\mathcal{L}_x$, which keeps $\Delta\mathbf{x}$ small and aligned with the on-manifold carrier direction, a constraint that *RandPerturb* and *Data Taggants* lack.

H.4 Scalability of Owner Attribution

We test attribution stability as the marketplace buyer size B varies. Figure 12 reports *Owner Attribution Accuracy (OAA, %)* at $B \in \{1, 2, 5, 10, 50, 100, 500, 1000\}$ across all five benchmarks, comparing five methods: **Ours-WB (white-box)**, **Ours-BB (black-box)**, *RandPerturb-WB*, *Backdoor*, and *Data Taggants*. Results are averaged over 30 random B-subsets of the 1000 private-key pool with one-standard-deviation bands. OAA is computed on the same 50 watermarked suspects from Table 1, under the fidelity budget $|\Delta\mathrm{Acc}| \leq 1\%$, where $\Delta\mathrm{Acc}$ denotes accuracy drop on benign test nodes.

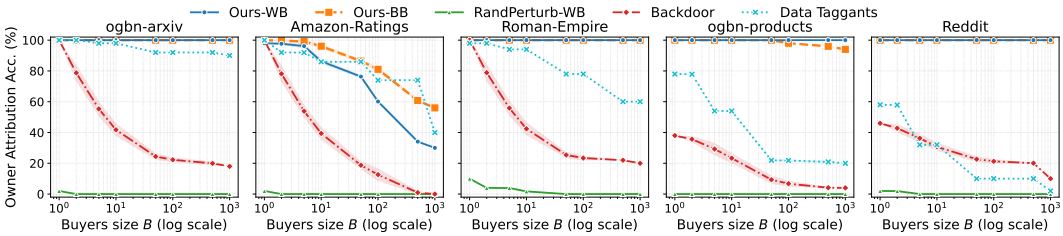


Figure 12: Owner Attribution Accuracy (%) versus the number of marketplace buyers B on five datasets. Each curve is the mean accuracy at each buyers size. Shaded band shows the standard deviation.

Ours-WB and Ours-BB sustain OAA near 100% across all B on ogbn-arxiv, Roman-Empire, ogbn-products, and Reddit, whereas every baseline decays as B grows. *Backdoor*'s cross-trigger interference drives OAA down to 0–20% as B grows, while *RandPerturb-WB* attributes at random ($\approx 0\%$ at $B = 1000$). *Data Taggants*, the strongest baseline, still declines from 90% on ogbn-arxiv to 2% on Reddit at $B = 1000$, as its per-buyer key signal is too weak to keep the owner separable once foreign buyers accumulate. On **Amazon-Ratings**, OAA degrades gracefully with B, reaching 30% (Ours-WB), 40% (Data Taggants), and 56% (Ours-BB) at $B = 1000$, rather than collapsing outright.

H.5 Cross-Probe Owner attribution ($B = 2$)

Our goal is a scalable owner attribution, but this is meaningful only when each candidate watermarked suspect model shows watermark signal on its own owner's private key (*own-fire*) and no signal on other owners' keys (*silent*). To test whether our method and the candidate baselines support scalable owner attribution, we first run a two-probe pilot in which each owner is assigned two triggers for the trigger-based methods *Backdoor*, *BkdOwn*, and *ImpWM*, or two private keys for the key-based methods Ours, *RandPerturb*, and *Data Taggants*, and we check whether the foreign probe induces an unexpected watermark signal (*cross-fire*).

Therefore, we conduct a cross-probe evaluation at $B=2$ owners, where each owner independently configures separate probes and trains separate suspect models; each probe is then applied to both models to test whether cross-fire occurs on non-matching suspects. And we define a threshold for each method to determine whether the probe fires on suspect models. For the trigger-based methods *Backdoor*, *BkdOwn*, and *ImpWM*, the threshold is the *Chance* level $1/C$ for C classes, where detection fires if the watermark signal (VSR) exceeds $1/C$. For the key-based methods Ours, *RandPerturb*, and *Data Taggants*, the threshold is $\kappa = \log_{10}(\tau/B) = \log_{10}(0.05/2) \approx -1.60$, where detection fires if the watermark signal $\log_{10}(p) < \kappa$.

Table 25 shows that the trigger-based methods *Backdoor*, *BkdOwn*, and *ImpWM* all exhibit unexpected cross-fire, where the foreign trigger fires on the non-matching suspect. *ImpWM* is the most extreme: its trigger fires at 100% on every suspect and even on the benign model, so it cannot separate owners at all. *BkdOwn*'s foreign signal matches its own signal, and *Backdoor* cross-fires most strongly on Amazon-Ratings, where the foreign trigger fires above the own trigger. In contrast, the key-based schemes attribute cleanly. Ours and the *Data Taggants* baseline fire only on their own key and keep silent on the foreign key and on the benign model, confirming that the keys are owner-specific. *RandPerturb* produces no signal on any key, consistent with it being a random-perturbation baseline that carries no recoverable watermark. This shows that cross-fire is intrinsic to trigger-based watermarks, whereas key-based schemes, including the *Data Taggants* baseline, support reliable owner attribution without interference from other owners' keys or benign models.

H.6 Runtime and computational cost

Beyond detection quality, deployment requires that the wall-clock cost of watermark injection and per-suspect verification remain acceptable as graph size grows. Injection is a one-time owner-side operation at release time; per-suspect verification is a repeated verifier-side operation at query time. We report each separately: *offline* injection and *online* pure verifier cost, excluding the adversary's suspect training which is outside the owner's control. Concretely, *offline injection* is the wall-clock to produce the released watermarked dataset from one owner-key; *online per-suspect verification* is the wall-clock from receiving a suspect model to producing the joint detection + attribution decision over the B-buyer key pool. For representative comparison we include one baseline per category:

Table 25: Owner attribution evaluation on two-probe settings. **Bold** cells mark the unexpected cross-fire.

Method	Dataset (probe)	Model	Watermark signal on a's probe	Watermark signal on b's probe	Threshold
RandPerturb	ogbn-arxiv, Owner a's private key	Suspect a	−1.05 (silent)	−0.15 (silent)	−1.60
	ogbn-arxiv, Owner b's private key	Suspect b	−0.01 (silent)	−0.76 (silent)	
	ogbn-arxiv, benign	Benign	0.00 (silent)	−0.18 (silent)	
	Amazon, Owner a's private key	Suspect a	−0.48 (silent)	−0.42 (silent)	−1.60
	Amazon, Owner b's private key	Suspect b	−0.59 (silent)	−0.70 (silent)	
	Amazon, benign	Benign	−0.54 (silent)	−0.50 (silent)	
Backdoor (Xu et al., 2023)	ogbn-arxiv, Owner a's trigger	Suspect a	37.4% (own-fire)	**10.7% (cross-fire)**	2.5%
	ogbn-arxiv, Owner b's trigger	Suspect b	1.3% (silent)	37.5% (own-fire)	
	ogbn-arxiv, benign	Benign	0.1% (silent)	0.1% (silent)	
	Amazon, Owner a's trigger	Suspect a	68.1% (own-fire)	1.6% (silent)	20.0%
	Amazon, Owner b's trigger	Suspect b	**34.0% (cross-fire)**	79.1% (own-fire)	
	Amazon, benign	Benign	8.3% (silent)	0.0% (silent)	
ImpWM (Zhang et al., 2024a)	ogbn-arxiv, Owner a's trigger	Suspect a	100% (own-fire)	**100% (cross-fire)**	2.5%
	ogbn-arxiv, Owner b's trigger	Suspect b	**100% (cross-fire)**	100% (own-fire)	
	ogbn-arxiv, benign	Benign	100% (fires)	100% (fires)	
	Amazon, Owner a's trigger	Suspect a	100% (own-fire)	**100% (cross-fire)**	20.0%
	Amazon, Owner b's trigger	Suspect b	**100% (cross-fire)**	100% (own-fire)	
	Amazon, benign	Benign	100% (fires)	100% (fires)	
BkdOwn (Li et al., 2023)	ogbn-arxiv, Owner a's trigger	Suspect a	98.2% (own-fire)	**99.0% (cross-fire)**	2.5%
	ogbn-arxiv, Owner b's trigger	Suspect b	**97.8% (cross-fire)**	98.9% (own-fire)	
	ogbn-arxiv, benign	Benign	0.0% (silent)	0.0% (silent)	
	Amazon, Owner a's trigger	Suspect a	100% (own-fire)	**100% (cross-fire)**	20.0%
	Amazon, Owner b's trigger	Suspect b	**100% (cross-fire)**	100% (own-fire)	
	Amazon, benign	Benign	0.0% (silent)	0.0% (silent)	
Data Taggants (Bouaziz et al., 2025)	ogbn-arxiv, Owner a's private key	Suspect a	−16.02 (own-fire)	−0.65 (silent)	−1.60
	ogbn-arxiv, Owner b's private key	Suspect b	0.00 (silent)	−16.02 (own-fire)	
	ogbn-arxiv, benign	Benign	0.00 (silent)	−0.65 (silent)	
	Amazon, Owner a's private key	Suspect a	−4.11 (own-fire)	0.00 (silent)	−1.60
	Amazon, Owner b's private key	Suspect b	−0.05 (silent)	−4.11 (own-fire)	
	Amazon, benign	Benign	−0.05 (silent)	−0.20 (silent)	
Ours	ogbn-arxiv, Owner a's private key	Suspect a	−37.10 (own-fire)	−0.11 (silent)	−1.60
	ogbn-arxiv, Owner b's private key	Suspect b	−0.50 (silent)	−25.51 (own-fire)	
	ogbn-arxiv, benign	Benign	−1.25 (silent)	−0.44 (silent)	
	Amazon, Owner a's private key	Suspect a	−4.35 (own-fire)	−0.84 (silent)	−1.60
	Amazon, Owner b's private key	Suspect b	−0.24 (silent)	−3.06 (own-fire)	
	Amazon, benign	Benign	−0.42 (silent)	−1.09 (silent)	

Table 26: Runtime of watermark injection and per-suspect verification over the B=1000-buyer key pool. *offline injection* and *online per-suspect verification* are mean ± std over 5 runs. Units: seconds.

Dataset	Method	Offline injection (s)	Online per-suspect verification (s)
Roman-Empire (small)	Ours-WB	102 ± 8	2 ± 0.3
	Ours-BB	102 ± 8	173 ± 20
	Backdoor (Xu et al., 2023)	3.2 ± 0.6	1 ± 0.2
	Data Taggants (Bouaziz et al., 2025)	85 ± 11	1 ± 0.2
ogbn-arxiv (medium)	Ours-WB	3188 ± 19	22 ± 2
	Ours-BB	3188 ± 19	291 ± 30
	Backdoor (Xu et al., 2023)	3.5 ± 0.7	5 ± 0.5
	Data Taggants (Bouaziz et al., 2025)	505 ± 24	2 ± 0.3
ogbn-products (large)	Ours-WB	5040 ± 180	200 ± 15
	Ours-BB	5040 ± 180	770 ± 37
	Backdoor (Xu et al., 2023)	4.1 ± 0.9	30 ± 3
	Data Taggants (Bouaziz et al., 2025)	893 ± 43	10 ± 1

Backdoor (Xu et al., 2023) for trigger-based methods and *Data Taggants* (Bouaziz et al., 2025) for private-key-based methods. BkdOwn (Li et al., 2023) and ImpWM (Zhang et al., 2024a), both trigger-based, are omitted. We measure runtime on three graph scales, Roman-Empire (small, 22K nodes), ogbn-arxiv (medium, 169K nodes), and ogbn-products (large, 2.4M nodes), to characterize how injection and verification costs scale with graph size.

Ours incurs a heavier wall-clock cost than baselines on both stages. **Offline injection**: Ours combines surrogate GCN training with a multi-regularizer embedding optimization over per-class subgraphs, while Backdoor completes injection via trivial data manipulation and Data Taggants uses a lighter single-loss gradient-matching routine, giving Ours a 10-15× overhead. Ours-WB and Ours-BB share the same offline injection pipeline; they diverge only at online verification. **Online per-suspect verification**: Ours-WB is dominated by the surrogate-suspect subspace alignment step, whose cost scales linearly with the number of nodes; trigger baselines query only the watermark-node subgraph and Data Taggants operates on isolated key vectors. Ours-BB additionally distills a student model from the suspect, which dominates its online cost. Despite this heavier cost, Ours delivers the best watermark detection and owner attribution accuracy across all datasets (§4.2).