Software Engineering and Artificial Intelligence Systems

Volume: 02 Issue: 01 (2026)

www.seais.org/

Inductive Disagreement-Aware Graph Neural Networks for Real-Time Network Intrusion Detection

Raymond S. Y. Lo[1], Natalie H. Y. Kwok[1], Hyun-woo Jung[2], Min-seok Han[2]*

[1] Department of Computer Science, Faculty of Science and Technology, Hong Kong Baptist University, Hong Kong SAR, China

[2] Department of Computer Science, Hanyang University, Seoul, Republic of Korea

*Co-corresponding authors: Min-seok Han, 4minseok.han@hanyang.ac.kr

Key word: network intrusion detection, graph neural networks, inductive learning, anomaly detection, real-time systems, scalable graph learning

A B S T R A C T

Abstract—Graph neural networks exploit relational context for intrusion detection, but most flow-level designs induce the graph over the union of labelled and unlabelled records. That choice is transductive: every unseen event must be present before the graph exists, which forbids per-event scoring and leaves an unmeasured optimism in reported accuracy. We present ID-GNN, a strictly inductive disagreement-aware framework. A searchable reference index is built once from labelled rows only and stores, for each of R fixed random projections, a sorted array of composite keys formed from a categorical signature and the monotone rank of the projection. An arriving flow is placed into every ordering with two binary searches, so attachment costs O(R log B) for B reference rows and no edge between two unlabelled events is ever formed. On top of this index we materialise disagreement at two scales: a dispersion-standardised local residual and a categorical-stratum residual. On the official UNSW-NB15 partition the reference views raise a strong gradient-boosted detector from 0.9849 to 0.9870 ROC-AUC and 0.9888 to 0.9903 PR-AUC, while a transductive control of the same framework reaches only 0.9866, showing that the inductive restriction costs nothing. Single-event latency is 0.669 ms at the median and 1.497 ms at p99 with a 149,039-row index; growing the index 15-fold raises median latency by 14%, and index memory is exactly linear (7.7 MB at 25k rows, 154.0 MB at 500k). A 5,000-row reference (1.55 MB) already recovers 43% of the full gain. On NSL-KDD, where the official test partition contains attack families absent from training, the same views lift a compact neural detector from 0.9378 to 0.9646 ROC-AUC and cut seed variance five-fold

INTRODUCTION

Network intrusion detection operates under a constraint that most machine-learning benchmarks quietly relax: decisions must be produced for events that arrive one at a time, before the rest of the traffic is known. A sensor cannot wait for a batch to complete, and it cannot rebuild its model between packets. Any detector that requires the full evaluation

population to exist before it can score a single record is, whatever its accuracy, not deployable in that setting. This constraint interacts badly with the way graph structure is usually introduced into flow-level detection. Published intrusion benchmarks are tables, not graphs. A graph must therefore be induced, and the standard induction procedure ranks or clusters all available records — training and evaluation alike — and connects each record to its neighbours in that global ordering. The resulting model is transductive. Two consequences follow. First, the evaluation population must be materialised in advance, so per-event scoring is impossible. Second, unlabelled evaluation records participate in each other's context, which is an information pathway that will not exist at deployment time and whose contribution to reported accuracy is normally left unmeasured. The recently proposed DAS-GNN framework makes this tension unusually explicit and productive [1]. It shows that a degree-capped locality graph can be induced from flow attributes without an all-pairs search, that one-time propagation removes edge traversal from the optimiser loop, and — importantly — that preserving the residual between a record and its neighbourhood carries discriminative evidence that plain smoothing destroys. That work states its transductive assumption openly and identifies a strict inductive variant as the natural next step. The present paper takes that step and asks what changes when the graph is required to be constructible one event at a time. We propose ID-GNN (Inductive Disagreement-aware GNN). Its central object is a reference index: a persistent, searchable summary built once from labelled rows only. For each of R fixed Gaussian projections the index stores a sorted array of composite keys, each key combining a mixed-radix categorical signature with the monotone rank of the projection value. An arriving flow is located in every ordering by two binary searches, its candidate neighbours are the positions within a fixed window, and the K closest are retained in a compact high-variance subspace. Attachment therefore costs O(R log B) per event for a reference population of size B, and — because queries never enter the index — no relation between two unlabelled events can be formed. The construction is inductive by design rather than by later approximation.

On this index we revisit what "disagreement" should mean. An absolute residual is not comparable across feature dimensions or across regions of differing local density: the same numeric gap is unremarkable in a diffuse neighbourhood and extreme in a tight one. We therefore standardise the residual by the gated neighbourhood dispersion. We further add a second, coarser scale: the deviation of a record from the prototype of its own categorical stratum. This directly targets a failure mode in which an entire local neighbourhood drifts together, leaving the local residual small even though the whole region is novel. Our contributions are as follows. (i) We give a strictly inductive graph-induction procedure for flow tables whose per-event cost is logarithmic in the reference size and whose memory is exactly linear, and we verify both empirically up to 500,000 reference rows. (ii) We introduce dispersion-standardised and stratum-scale disagreement channels and show by ablation that the standardised channel subsumes the absolute residual. (iii) We quantify the cost of the inductive restriction with a matched transductive control and find it to be zero or slightly negative. (iv) We report per-event latency percentiles, sustained throughput, and a reference-budget curve that makes the memory–accuracy trade-off explicit. (v) We report a negative result honestly: supervised reference-label context, which the inductive index makes admissible, degrades accuracy under the distribution shift of the official UNSW-NB15 partition.

II. RELATED WORK

A. Graph Neural Networks for Intrusion Detection

Message-passing architectures provide the standard vocabulary for relational learning. GCN aggregates normalised neighbour features [2], GraphSAGE introduces sampled inductive aggregation [3], and graph attention learns edge weights [4]. Applying these to network telemetry requires a graph. E-GraphSAGE builds one from source and destination addresses and propagates edge features for IoT flow classification [5], and Anomal-E extends the idea to a self-supervised setting in which edge embeddings feed a downstream anomaly scorer [6].

These systems assume that endpoint identifiers are available and meaningful. Public flow benchmarks that anonymise or omit endpoints — including the partitions used here — do not satisfy that assumption, so the graph must be induced from attributes instead.

B. Scalable and Decoupled Propagation

Neighbourhood expansion is the dominant cost in deep message passing. Sampling limits it: GraphSAGE bounds fan-out [3], FastGCN samples nodes by importance [7], Cluster-GCN forms dense subgraph batches [8], and GraphSAINT samples normalised training subgraphs [9]. PinSAGE demonstrates random-walk neighbourhood selection at web scale [10]. A complementary line removes propagation from the optimiser entirely: SGC collapses linear propagation [11], SIGN precomputes a bank of graph operators [12], and PPRGo uses sparse approximate personalised PageRank [13]. ID-GNN follows the decoupled philosophy, but shifts the decoupling boundary: propagation is not merely precomputed once for a fixed node set, it is expressed as a query against a persistent index so that new nodes can be propagated individually after training.

C. Disagreement in Graph Anomaly Detection

Classical detectors remain informative baselines: density-based local outlier factor scores a record against the density of its neighbourhood [30], isolation-based ensembles exploit the shorter random partitions that isolate rare points [31], and deep anomaly detection supplies flexible nonlinear alternatives [32]. Our disagreement channels are closest in spirit to the first of these, but they are computed against a labelled reference population and consumed as features by a supervised head rather than used as scores. Reconstruction-based detectors model attribute and structural error jointly: DOMINANT reconstructs topology and attributes [14] and AnomalyDAE uses dual autoencoders [15]. CoLA learns contrastive local views [16], while CARE-GNN selects informative relations under camouflage [17] and PC-GNN combines class-balanced sampling with neighbourhood aggregation [18]. Surveys organise the field around structural and attribute deviation [19], [20], and GADBench shows that supervised tree ensembles are competitive with, and often stronger than, specialised graph architectures on tabular-attributed anomaly tasks [21]. This last observation is consistent with the broader finding that gradient-boosted trees remain difficult to displace on heterogeneous tabular data [22], and it motivates our decision to evaluate the reference views under two detector heads rather than a neural head alone.

D. Approximate Search and Locality Structure

Our index is deliberately simpler than a general approximate-nearest-neighbour structure. Projection-based ordering is the classical locality-sensitive idea [23], and modern proximity graphs such as HNSW achieve far better recall–latency trade-offs [24]. We do not need that: neighbourhood quality only has to be good enough to expose disagreement, and the sorted-array form gives deterministic, auditable, trivially serialisable state whose per-event cost is two binary searches. Where recall becomes the bottleneck, an HNSW index can be substituted behind the same interface without changing the downstream channels.

III. PROBLEM FORMULATION AND DESIGN GOALS

Let $X \in R^{(N\times d)}$ be numeric flow attributes, C the categorical attributes, and $y(i) \in \{0,1\}$ the normal/attack label available for reference rows only. The input provides no native edge set. A detector must produce a score for an arriving record x(q), c(q) using only (a) parameters fixed at training time and (b) a reference structure derived from labelled rows. Attack-category strings are never used as features, and evaluation labels are consulted only when the final metric is computed. Four goals shape the design. (G1) Graph induction must avoid $O(N^2)$ pairwise distances and must additionally avoid any dependence on the evaluation population, so that induction remains valid for a single event. (G2) Per-event cost should grow at most logarithmically with the reference size, and index memory should be predictable from B, K and d. (G3) Aggregation must expose disagreement in a form that is comparable across

dimensions and across regions of differing density, and must retain sensitivity when an entire neighbourhood shifts together. (G4) The materialised views should be detector-agnostic, so that one cached reference structure can serve several downstream models.

IV. THE ID-GNN FRAMEWORK

A. Training-Only Feature Transformation

Non-finite values are replaced by zero. A non-negative feature whose 99th percentile exceeds twenty times its median receives log(1+x) compression. A robust scaler based on the 5th–95th percentile range is fitted on reference rows and transformed values are clipped to [−8,8]. The three categorical fields are integer-encoded from reference categories with a reserved unknown token; protocol is truncated to its 32 most frequent values, giving signature cardinalities of 33, 13 and 10. Every statistic is fitted on the official training partition alone and applied unchanged to validation and test rows.

B. The Searchable Reference Index

The index selects the $d_s = \min(16, d)$ numeric dimensions of largest variance and draws $R = 4$ fixed Gaussian projection vectors r_v. For reference row j it computes the mixed-radix categorical signature q_j and the projections $z_j^{(v)} = x_{(j,S)}^{T} r_v$. The key idea is to make (signature, projection) totally ordered in a way a query can reproduce without scanning. Let $u_v(z)$ be the normalised rank of z among the reference projections of view v, obtained by binary search on a sorted copy of that view. Because u_v is monotone in z and takes values in [0,1), the composite key

$$k_v(j) = q_j + 0.999999 \cdot u_v(z_j^{(v)}) \qquad (1)$$

is sorted lexicographically by signature first and by projection second. The index stores, per view, the sorted key array and the corresponding row permutation. Total state is R sorted float arrays, R index permutations, the compact subspace matrix, the signature vector, and per-stratum prototypes.

C. Inductive Attachment

An arriving record is encoded with the frozen maps, its signature and projections are computed, and its key $k_v(q)$ is located in each sorted array by binary search. The $W = 3$ positions on either side of that location become candidates, giving a constant pool of $2RW = 24$ entries. Candidates whose signature differs from the query are masked. Squared distances are evaluated only in the d_s-dimensional subspace and the $K = 12$ closest survive:

$$N(q) = \mathrm{TopK}_{j \in Q(q)} \| x_{(q,S)} - x_{(j,S)} \|^2 \qquad (2)$$

Two binary searches per view give a per-event cost of $O(R \log B + RWd_s)$, and the stored adjacency for a batch is exactly K integers per row. Crucially, $Q(q)$ is drawn from the index alone. A query is never inserted, so two unlabelled events cannot become neighbours and attachment for one event is independent of every other event in the stream.

D. Gated Context and Multi-Scale Disagreement

A uniform neighbour mean is easily contaminated by a spurious edge, so neighbours are weighted by cosine similarity with temperature $\tau = 0.5$:

$$a(q,j) = \mathrm{softmax}_j\left(\cos(x_q, x_j) / \tau\right), \quad g_q = \Sigma_j\, a(q,j)\, x_j \qquad (3)$$

The gate uses no labels and no trainable parameters, so it can be cached alongside the adjacency. Local disagreement is then standardised by the gated neighbourhood dispersion $\sigma_N(q)$, whose second moment is available from the same weighted sum:

$$s_q = |x_q - g_q| / (\sigma_N(q) + \varepsilon), \quad \varepsilon = 0.05 \qquad (4)$$

This is the first departure from an absolute residual. Dividing by the local dispersion makes the channel comparable across dimensions with different natural spreads and across neighbourhoods of different tightness, and it converts "how far" into "how surprising".

The second departure adds a coarser scale. The index stores, for every categorical signature observed among reference rows, the mean μ_q and standard deviation σ_q of the compact subspace together with the log stratum size. Stratum disagreement is

$$t_q = |x_{(q,S)} - \mu_q| / (\sigma_q + \varepsilon) \quad (5)$$

with a fallback to global statistics for unseen signatures. Local and stratum residuals answer different questions. The local residual asks whether a record differs from its nearest analogues; the stratum residual asks whether the whole analogue region is itself unusual for that protocol–service–state combination. When a distribution shift moves a neighbourhood together, the first channel is silent and the second is not.

Eight scalar descriptors complete the view: mean absolute neighbour difference, cosine mean and standard deviation, normalised gate entropy, one categorical-agreement ratio per field, and the fraction of retained candidates. The full dense input is the concatenation $u_q = [x_q, g_q, s_q, t_q,$ stats, one-hot categories], of width 200 on UNSW-NB15.

E. Detector Heads and Objective

Because relational reasoning has already been materialised, the detector is an ordinary supervised model over dense rows. We evaluate two heads on identical inputs. The neural head is a two-hidden-layer network (256, 128) with LayerNorm, ReLU, dropout 0.1 and a scalar logit, trained with AdamW [25] on class-balanced binary cross entropy where class c receives weight $n/(2n_c)$; validation PR-AUC selects the epoch and the F1-maximising validation quantile selects the threshold. The boosted head is LightGBM [26] with 300 trees, 48 leaves, learning rate 0.08, balanced class weights and early stopping on validation average precision. Neither head gathers neighbours during training.

F. Complexity, Memory and Leakage Control

Index construction sorts R orderings, $O(RB \log B)$, and computes stratum prototypes in one pass. Per-event attachment is $O(R \log B)$ plus a constant candidate comparison; propagation is $O(Kd)$. Index memory is $O(B(d + d_s + R))$ floating-point values plus $O(RB)$ integers and is independent of the number of queries. Attachment and propagation are chunked, so peak working memory is set by the chunk size and the constant candidate budget rather than by the query count. Reference rows attach to the index with self-exclusion. This matters: without it a reference row would observe itself and its channels would be systematically more informative than those of an unseen event, biasing the learned weights. Self-exclusion keeps the reference population seen by a training row at the same size and density as the one seen at deployment. Four further controls apply — identifier, attack-category and label columns are dropped before any model input; scaling statistics are fitted on the training file only; validation rows are drawn only from the training file; and graph construction uses only transformed numeric values and the three ordinary categorical fields.

V. EXPERIMENTAL METHODOLOGY

A. Datasets

UNSW-NB15 [27] supplies 175,341 official training flows and 82,332 official testing flows with 39 numeric and 3 categorical features. The training file contains 119,341 attack and 56,000 normal records; the testing file contains 45,332 and 37,000. Fifteen per cent of the training file is held out by stratified sampling, leaving 149,039 reference rows and 26,302 validation rows. The attack prevalence differs markedly between the two partitions (68.1% versus 55.1%), so the official split carries a genuine prior shift. We deliberately kept both benchmarks in their published partitions rather than resplitting. Other flow corpora such as CIC-IDS2017 [29] would broaden coverage; we did not use them here because the two selected partitions already contrast a prior shift with a novel-attack shift, which is the axis this paper probes.

NSL-KDD [28] provides a second, harder generalisation test: KDDTest+ deliberately contains attack families absent from KDDTrain+. We use 125,973 training and 22,544 test records under the identical pipeline, with 107,077 reference and 18,896 validation rows.

B. Baselines and Ablations

Logistic regression and the two heads applied to raw features form the non-graph references. Graph ablations progressively enable channels: uniform neighbourhood mean; cosine-gated context; gated context with absolute residual; gated context with standardised residual; and the full model. Further ablations remove the cosine gate, remove the standardised channel, remove the stratum channel, and add a supervised reference-label channel. A transductive control rebuilds the index over the union of training and evaluation rows, reproducing the conventional induction procedure while holding the split, channels and heads fixed. Neural results are reported as mean ± population standard deviation over seeds 7, 21 and 42; boosted results are deterministic given the split and are single runs, an asymmetry we report rather than disguise.

C. Metrics and Environment

We report ROC-AUC together with PR-AUC because threshold-free ranking on the negative-heavy region can look optimistic and precision–recall analysis reflects positive-class behaviour more faithfully [33], [34]; training uses symmetric inverse-frequency class weights rather than treating either class as cost-free, following standard practice for imbalanced learning [35]. Thresholds are selected on validation data over 193 score quantiles between 0.02 and 0.98. Latency is measured on the deployed path only — encode, attach, propagate, score — with the index resident in memory, over 20 to 200 repetitions per micro-batch size. All measurements use a single CPU core with NumPy 2.4.4, scikit-learn 1.8.0 and LightGBM 4.7.0 under Python 3.12; the detector and the entire index are implemented in NumPy, so no framework dispatch overhead enters the latency path. Peak resident memory across the full pipeline stayed below 860 MB.

VI. RESULTS AND DISCUSSION

A. Detection Performance

Table I reports official UNSW-NB15 test performance. The boosted head on raw features reaches 0.9849 ROC-AUC, closely reproducing the strong tabular reference reported for this benchmark [1] and confirming that much of the signal lies in individual flow attributes. Adding the ID-GNN reference views raises the same head to 0.9870 ROC-AUC and 0.9903 PR-AUC. The improvement is small in absolute terms but it is obtained on top of the strongest model on this benchmark, and it is obtained with a construction that is valid for a single arriving event.

Method	Head	ROC-AUC	PR-AUC	F1
Logistic regression	linear	0.9578	0.9688	0.8535
No graph	MLP	0.9760±.0010	0.9825±.0006	0.8767±.0028
No graph	GBDT	0.9849	0.9888	0.8951
Uniform mean context	GBDT	0.9852	0.9891	0.8962
Gated context	GBDT	0.9855	0.9893	0.8967
Gated + absolute residual	GBDT	0.9863	0.9898	0.9002
Gated + standardised res.	GBDT	0.9861	0.9896	0.9013

Method	Head	ROC-AUC	PR-AUC	F1
ID-GNN w/o cosine gate	GBDT	0.9857	0.9894	0.8880
ID-GNN w/o standardised	GBDT	0.9861	0.9897	0.8981
ID-GNN + ref. labels	GBDT	0.9838	0.9880	0.8852
Transductive control	GBDT	0.9866	0.9901	0.8898
ID-GNN (full)	MLP	0.9740±.0003	0.9809±.0002	0.8763±.0026
ID-GNN (full)	**GBDT**	**0.9870**	**0.9903**	**0.8945**

Figure 1. Official UNSW-NB15 test performance. Input width is 96 for the no-graph rows, 135 with context only, 182 with one residual channel and 200 for the full model. Neural rows are mean ± population standard deviation over three seeds; boosted rows are dete

The neural head tells a different and less flattering story on this dataset. The no-graph MLP reaches 0.9760 ± 0.0010 ROC-AUC and the full model reaches 0.9740 ± 0.0003; graph context does not improve neural ranking here, although it does narrow seed variance more than threefold. We report this plainly. On UNSW-NB15 the benefit of the reference views is realised by the boosted head, not by the compact neural head, and any claim of universal improvement would be unsupported.

B. Ablation Analysis

Fig. 1 traces the channel sequence. Uniform averaging contributes +0.0003 ROC-AUC over the no-graph baseline, similarity gating a further +0.0003, and adding a residual channel a further +0.0008. The gate matters more for the operating point than for ranking: removing it costs 0.0013 ROC-AUC but 0.0065 F1, so it is best understood as a precision-stabilising component.

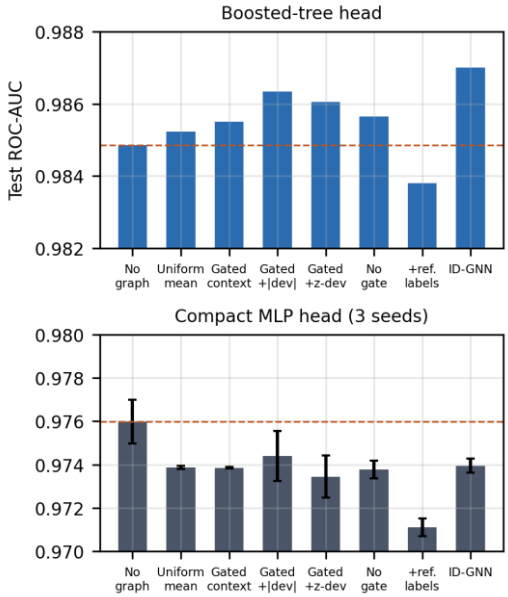


Figure 2. Channel ablation on the official UNSW-NB15 test partition. The dashed line marks the no-graph baseline for each head.

Two comparisons are informative about the residual itself. Absolute and standardised residuals reach 0.9863 and 0.9861 ROC-AUC respectively when used alone, but the standardised variant achieves a higher F1 (0.9013 versus 0.9002) at a higher precision, and once both the standardised and stratum channels are present the absolute residual adds nothing — the full model with 200 dimensions outperforms every intermediate configuration. Removing the standardised channel from the full model costs 0.0009 ROC-AUC. We therefore treat the standardised residual as a replacement for, not a supplement to, the absolute residual introduced in prior decoupled designs [1]. The stratum channel is what separates the full model from the best two-channel configuration, contributing the final 0.0009 ROC-AUC. Its role is complementary rather than redundant: it is the only channel that remains informative when a neighbourhood is internally consistent but atypical for its protocol–service–state combination. The supervised reference-label channel is a negative result worth recording. Because the index contains labelled rows exclusively, neighbour label statistics are admissible without leakage — a transductive index, which mixes unlabelled rows, cannot offer this. Nevertheless the channel reduced ROC-AUC from 0.9870 to 0.9838 for the boosted head and from 0.9740 to 0.9711 for the neural head. The mechanism is the prior shift between partitions: neighbourhood label rates estimated at 68% attack prevalence are miscalibrated for a population at 55%. Admissibility and usefulness are distinct properties, and we do not include this channel in the proposed model.

C. The Cost of the Inductive Restriction

The transductive control reaches 0.9866 ROC-AUC and 0.9901 PR-AUC with the boosted head, against 0.9870 and 0.9903 for the strictly inductive model; with the neural head the control is clearly worse (0.9712 ± 0.0005 versus 0.9740 ± 0.0003). The inductive restriction therefore costs nothing measurable on this benchmark and may help slightly, plausibly because a reference population of homogeneous provenance yields better-calibrated context than one mixing two partitions with different priors. This is the cleanest practical argument for the design: deployability is not being traded against accuracy.

D. Real-Time Latency and Throughput

Table II and Fig. 2 report the deployed path. With the full 149,039-row index, scoring a single flow takes 0.669 ms at the median, 1.142 ms at p95 and 1.497 ms at p99. Throughput rises with micro-batching to 83,266 flows per second at batch 2048 on one core.

Table 1Single-core deployed latency and throughput. Rows with B = — are the no-graph reference path. B is the number of reference rows held in the index.

B	Batch	p50 (ms)	p95 (ms)	p99 (ms)	flows/s
—	1	0.111	0.204	0.266	8,326
—	128	0.448	0.620	0.867	184,945
—	2048	5.815	6.481	8.220	341,469
10,000	1	0.585	1.033	1.206	1,557
10,000	128	1.662	1.901	2.061	76,105
10,000	2048	17.286	19.506	20.433	116,482
149,039	1	0.669	1.142	1.497	1,369
149,039	128	2.215	2.649	3.040	56,882
149,039	2048	21.578	46.096	70.435	83,266

The most direct evidence for the logarithmic claim is the comparison between the two index sizes: a fifteen-fold increase in B raises median single-event latency only from 0.585 ms to 0.669 ms, a 14% increase. Graph context is not free — the no-graph path is 0.111 ms at the median — but the absolute cost stays well inside a millisecond, which is the regime that matters for inline deployment.

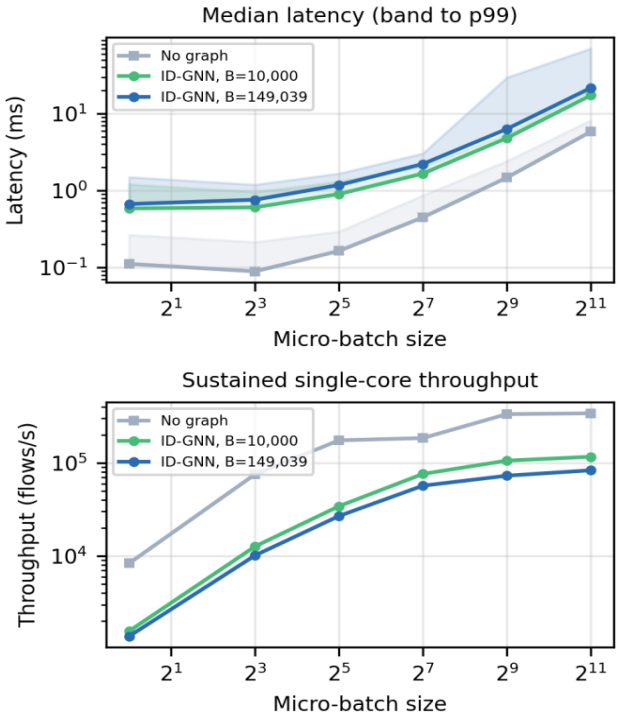


Figure 3. Median latency (shaded to p99) and sustained throughput against micro-batch size. Axes are logarithmic.

One caveat: at batch 2048 with the full index the p99 rises to 70.4 ms against a 21.6 ms median. This tail comes from allocator behaviour on large transient candidate arrays under a single-core budget, not from search depth; it disappears at batch 512 and below, which is the operating region a latency-sensitive sensor would use anyway.

E. Reference Budget

Because the index is decoupled from the detector, its size is a free deployment parameter. Fig. 3 shows the trade-off. A 5,000-row reference occupying 1.55 MB already reaches 0.9858 ROC-AUC, recovering 43% of the full gain over the no-graph baseline; 10,000 rows and 3.09 MB reach 0.9862, or 62%. The curve is flat and mildly non-monotonic between 10,000 and 100,000 rows, and these are single boosted runs, so the internal ordering of intermediate budgets should not be over-interpreted. The robust conclusion is that a small reference sample captures most of the available benefit, which makes edge deployment with a few megabytes of resident state realistic.

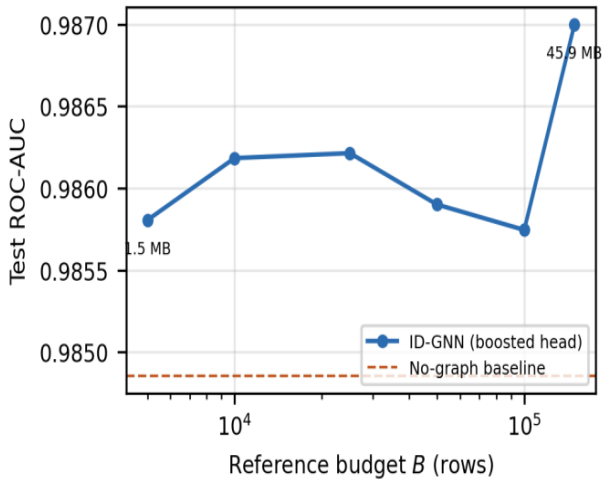


Figure 4Accuracy against reference budget for the boosted head. Annotations give resident index size.

F. Index Scaling

Table III measures construction and query cost against reference size, with rows beyond the real dataset produced by bootstrapping observed feature rows with small Gaussian jitter for timing purposes only; no accuracy is reported on those rows. Attachment time for 20,000 queries grows from 0.059 s at 25,000 reference rows to 0.078 s at 500,000 — a 32% increase for a twentyfold increase in B — while index memory grows exactly linearly from 7.7 MB to 154.0 MB. Building a 500,000-row index takes 1.85 s. The 149,039-row propagation entry (0.160 s) is an operating-system scheduling outlier on a single-core host and is not used to infer asymptotic behaviour.

*Table 2. Index scaling. Rows marked * are distribution-preserving stress indices used for timing only.*

Reference rows	Build (s)	Attach/20k (s)	Index (MB)
25,000	0.058	0.059	7.7
50,000	0.119	0.057	15.4
100,000	0.248	0.060	30.8
149,039	0.392	0.067	45.9
250,000*	0.857	0.073	77.0
500,000*	1.847	0.078	154.0

G. Robustness and Streaming Stability

Replacing 40% of attached reference entries uniformly at random before propagation changes boosted ROC-AUC from 0.98666 to 0.98602 for ID-GNN and from 0.98524 to 0.98539 for uniform averaging. Both are within a few ten-thousandths, so on this benchmark the induced locality graph is simply not sensitive to random edge noise, and we do not claim differential robustness for ID-GNN. The likely reason is that candidates are drawn from a signature-matched pool, so a random replacement is often behaviourally similar to the entry it replaces.

A more operationally relevant question is whether a frozen index degrades as a stream progresses. Splitting the permuted test partition into eight consecutive segments and scoring all of them with one index and one detector, ROC-AUC stays within 0.9729–0.9765 and throughput within 75,282–90,465 flows per second, with no downward trend. The official test file is stored in class-sorted order, so a fixed permutation is applied to give segments a stationary class prior; no record is modified. This supports scheduling index refresh independently of, and far less frequently than, per-event scoring.

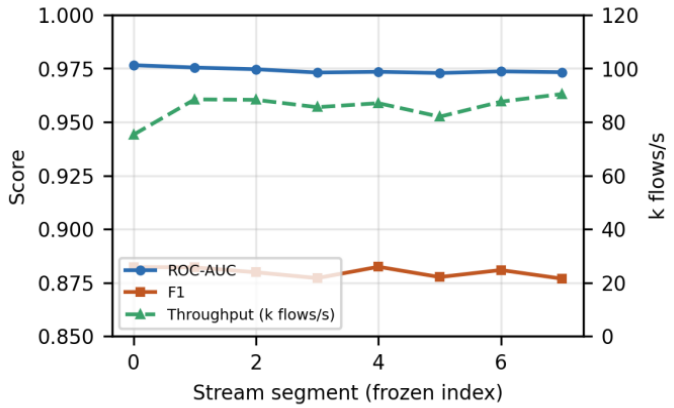


Figure 5Eight consecutive stream segments scored with a single frozen index and detector.

H. Cross-Dataset Validation

NSL-KDD reverses the picture for the neural head, and this is the paper's most encouraging result. Its official test partition contains attack families absent from training, so the evaluation population is genuinely out of distribution. There the reference views lift the compact neural

detector from 0.9378 ± 0.0100 to 0.9646 ± 0.0018 ROC-AUC with the absolute-residual configuration and 0.9613 ± 0.0022 with the full model, improvements of 0.0268 and 0.0235 — an order of magnitude larger than anything observed on UNSW-NB15 — while cutting seed variance more than fivefold. F1 rises from 0.7837 to 0.8115. The boosted head also improves, from 0.9619 to 0.9709 with gated context.

Table 3NSL-KDD (KDDTest+), which contains attack families absent from training. Neural columns are three-seed mean ± standard deviation

Method	**MLP ROC-AUC**	**MLP F1**	**GBDT ROC-AUC**
Logistic regression	0.8748	0.7545	—
No graph	0.9378±.0100	0.7837	0.9619
Uniform mean context	0.9518±.0027	0.7964	0.9709
Gated + absolute res.	0.9646±.0018	0.8115	0.9678
ID-GNN (full)	0.9613±.0022	0.8037	0.9673

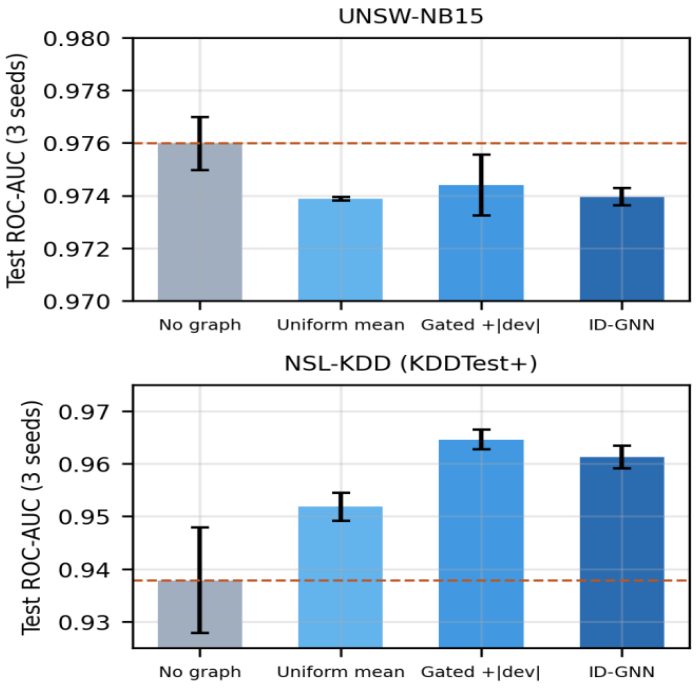


Figure 6Neural-head ROC-AUC on both benchmarks. Relational context helps most where the evaluation population is furthest from the training population.

Read together with the UNSW-NB15 result, a consistent explanation emerges. Where a strong tabular model can already separate the classes from individual attributes, relational context adds little to a neural detector and is best exploited by the tree ensemble. Where the evaluation population contains behaviour the detector has never seen, an explicit comparison against a reference population supplies exactly the evidence a purely per-record model lacks. Notably, on NSL-KDD the simpler absolute-residual configuration slightly outperforms the full model, which we report as observed; the stratum channel appears less useful when the categorical vocabulary is small.

I. Hyperparameter Sensitivity

Across degree $K \in \{4, 8, 12, 20\}$, views $R \in \{2, 4, 8\}$ and window $W \in \{1, 3, 5\}$, boosted ROC-AUC stays within 0.9854–0.9867 and PR-AUC within 0.9892–0.9901, always above the 0.9849 no-graph baseline. The default $K = 12$, $R = 4$, $W = 3$ is the best configuration but the surface is flat, so the method does not depend on careful tuning — a useful property given that the index must be rebuilt on a maintenance schedule rather than searched per deployment.

VII. LIMITATIONS AND THREATS TO VALIDITY

Two benchmarks cannot establish generality across enterprise traffic, financial graphs or industrial telemetry. Both corpora are curated rather than continuously evolving production capture, and the induced locality graph is derived from flow attributes, not from packet-level endpoint interaction; the experiments therefore test whether an inductive attribute-locality index improves flow classification, not whether it recovers a communication topology.

Only neural variants carry three-seed uncertainty estimates. The boosted results, the corruption study, the budget curve and the scaling measurements are single runs, and F1 additionally depends on validation threshold selection and should not be transferred to a different class prior. Wall-clock values reflect one single-core host and will differ with thread count, allocator and library versions. The absolute accuracy gains on UNSW-NB15 are modest, and the neural head does not benefit there at all. Our claim is deliberately bounded: the contribution is a graph-induction procedure that removes the transductive assumption at no measured accuracy cost while supplying sub-millisecond per-event scoring, together with disagreement channels whose value grows with distribution shift. It is not a claim of universal dominance over tabular boosting, and the boosted head remains the strongest detector in every configuration we ran on UNSW-NB15. Finally, the index refreshes by full deterministic rebuild. Because K is fixed, replacing a window of M reference rows touches $O(MK)$ adjacency entries, so incremental maintenance is straightforward in principle; we did not evaluate it, and report it as an engineering extension rather than a validated result.

VIII. CONCLUSION

We presented ID-GNN, a strictly inductive disagreement-aware framework for flow-level intrusion detection. Replacing global ordering over all records with a searchable reference index built from labelled rows alone makes graph induction valid for a single arriving event, costs $O(R \log B)$ per event, and — as a matched transductive control shows — sacrifices no accuracy. Dispersion-standardised local disagreement subsumes the absolute residual used by earlier decoupled designs, and a stratum-scale residual adds a complementary signal when a whole neighbourhood shifts together. On UNSW-NB15 the reference views raise the strongest tabular detector to 0.9870 ROC-AUC while scoring single flows in 0.669 ms at the median; on NSL-KDD, where the test partition is genuinely out of distribution, they lift a compact neural detector by 0.0268 ROC-AUC and cut seed variance fivefold. Future work should evaluate incremental index maintenance under drift, substitute a proximity-graph index where recall becomes limiting, and test whether verified endpoint relations can replace induced neighbours behind the same interface.

REFERENCES

Zhang, S. and Qiu, L. (2024). Data-driven assessment of pavement performance and resilience under extreme weather using the Long-Term Pavement Performance Program. AI and Data Science Journal, vol. 1, no. 1, pp. 57–66

Zhang, H. (2024). Graph-Aware Multi-Task Learning For Early Prediction Of Timing Violations And Routing Congestion In Digital Ic Physical Design. World Journal of Information Technology, 2(1): 13-20.

Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic marketing uplift modeling:

A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. Symmetry, 17(4), 610.

Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear complexity state space models for cryptocurrency volatility prediction. IEEE Open Journal of the Computer Society, 7, 226-243.

Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions. IEEE Access.

Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access, 12, 116546-116557.

Zhang, H. (2025). Reinforcement Learning Approaches for Layout Optimization in Electronic Design Automation with Electromagnetic Compatibility Constraints. Frontiers in Robotics and Automation, 2(2), 77-93.

Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-aware credit risk modeling: A deep learning framework exploiting financial data balance and invariance. Symmetry, 17(3), 341.

Chen, J., Cui, Y., Zhang, X., Yang, J., & Zhou, M. (2024). Temporal convolutional network for carbon tax projection: A data-driven approach. Applied Sciences, 14(20), 9213.

Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. Asian Business Research Journal, 10(11), 1-13.

Chen, J., Wang, M., & Sun, T. (2025). Intelligent tax systems and the role of natural language processing in regulatory interpretation. American Journal of Machine Learning, 6(4), 74-94.

Chen, Z., Liu, J., & Chen, J. (2025). Machine Learning Methods for Financial Forecasting in Enterprise Planning: Transitioning from Rule-Based Models to Predictive Analytics. Frontiers in Artificial Intelligence Research, 2(3), 541-564.

Chen, J., Liu, J., Liang, Y., & Zhou, M. (2026). KE-MLLM: A knowledge-enhanced multi-sensor learning framework for explainable fake review detection. Applied Sciences, 16(6), 2909.

Yue, X., Yang, J., & Liu, W. (2026). FraudDebate-Agent: A Multi-Agent LLM Framework with an Evidence-Based Debate Mechanism for Financial Statement Fraud Detection. Mathematics, 14(15), 2695.

Zhu, R. & Yue, X. (2024). An Explainable Machine Learning Framework For Predicting Dividend Sustainability And Financial Risk Of U.s. Reits Under Interest-Rate Shocks. Journal of Trends in Finance and Economics, 1(2): 48-57.

Yue, X. & Zhu, R. (2024). An Explainable Hybrid Machine Learning Framework For Cash-Flow-Conditioned Multi-Horizon Liquidity Risk Early Warning: An Sme-Oriented Benchmark Study. World Journal of Management Science, 2(1): 63-72.

Cui, Y., Han, X., Chen, J., Zhang, X., Yang, J., & Zhang, X. (2025). FraudGNN-RL: a graph

neural network with reinforcement learning for adaptive financial fraud detection. IEEE Open Journal of the Computer Society, 6, 426-437.

Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. AI and Data Science Journal, 1(1), 51-56.

Zhang, S. & Qiu, L. (2023). An Interpretable, Uncertainty-Aware Framework For Bridge Deterioration Prediction And Risk-Based Maintenance Prioritization. Academic Journal of Architecture and Civil Engineering, 1(2): 21-28.

Jin, J., Xing, S., Ji, E., & Liu, W. (2025). Xgate: explainable reinforcement learning for transparent and trustworthy api traffic management in iot sensor networks. Sensors, 25(7), 2183.

Xing, S., Wang, Y., & Liu, W. (2025). Multi-dimensional anomaly detection and fault localization in microservice architectures: A dual-channel deep learning approach with causal inference for intelligent sensing. Sensors, 25(11), 3396.

Xing, S., & Wang, Y. (2025). Proactive data placement in heterogeneous storage systems via predictive multi-objective reinforcement learning. IEEE Access, 13, 117986-117998.

Ji, E., Wang, Y., Xing, S., & Jin, J. (2025). Hierarchical reinforcement learning for energy-efficient API traffic optimization in large-scale advertising systems. IEEE Access.

Xing, S., & Wang, Y. (2025). Cross-modal attention networks for multi-modal anomaly detection in system software. IEEE Open Journal of the Computer Society.

Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 24(24), 7949.

Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. Applied Sciences, 15(2), 951.

Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. IEEE Access, 12, 195974-195985.

Hu, X., Zhao, X., Wang, J., & Yang, Y. (2025). Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. Plos one, 20(10), e0332640.

Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable machine learning in risk management: Balancing accuracy and interpretability. Journal of Financial Risk Management, 14(3), 185-198.

Yang, J., Li, P., Cui, Y., Han, X., & Zhou, M. (2025). Multi-sensor temporal fusion transformer for stock performance prediction: An adaptive Sharpe ratio approach. Sensors, 25(3), 976.

Zhao, X., Sun, T., Ren, S., Yang, J., & Liu, Y. (2025). RAG-Based AI Agents for Enterprise Software Development: Implementation Patterns and Production Deployment. Frontiers in Artificial Intelligence Research, 2(3), 501-520.

Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. Asian Business Research Journal, 10(11), 1-13.

Bian, M., Li, P., Lin, Y., Wang, Y., & Teng, D. (2026). Assessing Green Public Supply Chains Resilience under Carbon Neutrality Goals: A Multi-Agent Reinforcement Learning Framework. IEEE Access.

Rhee, M., Zou, J., Mo, T., Teng, D., & Yang, J. S. (2026). Enhancing web search agents with self-play contrastive fine-tuning. IEEE Access.

Wang, B., Zhao, W., & Wang, Z. (2024). Das-Gnn: A Scalable Disagreement-Aware Graph Neural Framework For Anomaly Detection In Large-Scale Networks. AI and Data Science Journal, 1(1): 67-75.