Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# When does graph structure help? A leakage-free controlled evaluation of heterogeneous graph learning for supply chain prediction

**Abu Bakkar Siddiq**
The University of Texas at Dallas
**Golam Sarwar Khondoker**
State University of Bangladesh
**Pelob Chakraborti**
Sonali Bank PLC
**Md Shaikh Kamran**
SSL Wireless
**Md Abdul Motin**
Janata Bank PLC
**Md Zulficar Hasan Joy**
Daffodil Institute of Information Technology
**Mahiya Watid**
Jahangirnagar University
**Mohammad Imtiaz Faisal**
imtiaz.faisal03@northsouth.edu

North South University

**Research Article**

**Keywords:** Graph neural networks, Evaluation methodology, Data leakage, Supply chain analytics, Negative results, Controlled ablation

**Posted Date:** November 23rd, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11491489/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

**Additional Declarations:** No competing interests reported.

## Abstract

Graph neural networks are increasingly applied to supply chain prediction on the reasoning that supply chains are relational systems. Most supply chain data, however, arrive as transactional tables from which a graph must be constructed by the analyst, and that construction step is rarely evaluated. We ask whether constructed relational structure adds predictive value beyond entity-level historical features, and under what conditions. Node features and graph edges are built from a historical window and labels from a strictly later window, with assertions enforcing the separation. Conventional models and heterogeneous graph neural networks receive identical features and training data, and the conventional comparator is selected on validation rather than test performance. Three architectures, eleven baselines, three datasets and nine tasks are evaluated over five seeds by five folds with corrected resampled inference. Controlled ablations hold architecture constant while varying edge content, including a degree-preserving randomisation of the real graph. No graph model achieved a statistically distinguishable advantage over the validation-selected comparator on any task. Message passing contributed significantly on the single dataset whose target-node projection carries above-chance neighbourhood signal, outperforming a degree-preserving randomisation by 0.056 AUC and a depth-matched self-loop control by 0.045, with effect sizes between 1.18 and 2.18. On two near-complete projections and one near-empty projection no such benefit is detectable. Topology diagnostics computed before any model is trained provide a plausible account of which case applies, suggesting that graph construction quality, rather than aggregation architecture, is the binding factor in these datasets.


## 1. Introduction

Supply chains are networks. Products belong to categories, customers purchase across ranges, sellers share carriers and warehouses, and disruptions propagate along these connections. Graph neural networks (GNNs) learn on such structure, and a growing literature applies them to supply chain planning, risk and link prediction [1–3]. The reasoning is intuitive: if the domain is relational, a relational model should help.

Most supply chain datasets, however, are not distributed as graphs. They are transactional tables, and the graph must be constructed by the analyst, who chooses node types, relations and thresholds. This construction step is rarely evaluated. A reported gain is usually attributed to the model rather than to the particular graph that was built, and almost never compared against the same architecture with message passing disabled.

Evaluation practice compounds the problem. Leakage — the introduction of information about the prediction target that would not legitimately be available at prediction time — was formalised by Kaufman et al. [4], who identify temporal leakage as among the most common and least visible forms. When node features and labels are aggregated over the same period, a feature such as a product's historical late-delivery rate and a label defined as that rate exceeding a median are near-identical quantities. In our own earlier work on this problem that overlap produced an apparent 21 per cent relative gain in F1 for a heterogeneous GraphSAGE model over the strongest tree-based baseline (0.924 against 0.763). Under the protocol described here the same architecture reaches AUC 0.696 against a validation-selected comparator at 0.779. This failure mode is subtle and reproducible, and illustrates why temporal separation should be made explicit in graph-learning evaluations.

This paper therefore asks three narrower questions than "do GNNs work for supply chains":

**RQ1.** Under a leakage-free temporal protocol with matched features, do heterogeneous GNNs improve on conventional machine learning baselines?

**RQ2.** Does relational message passing improve on the same architecture without it, holding network depth and capacity approximately constant?

**RQ3.** Does the predictive value of the graph relate to measurable properties of the constructed topology?

The contributions are a leakage-free temporal construction and evaluation protocol with explicit assertions separating feature and label windows; a matched comparison in which conventional baselines and GNNs receive identical features and training

data; controlled structure ablations that separate message passing from network depth, including a degree-preserving randomisation of the real graph; and topology diagnostics relating density, homophily and neighbourhood informativeness to observed benefit.

## 2. Literature review

# 2.1 Machine learning in supply chain analytics

Machine learning has been applied to demand forecasting, inventory optimisation and risk management for decades [5–6]. Ensemble methods, particularly random forests [7] and gradient boosting [8–9], remain strong on tabular supply chain data [10], consistent with broader evidence that tree ensembles remain competitive with deep architectures on tabular problems [11]. Deep learning has been applied to time-series forecasting and inventory problems [12]. Fraud and anomaly detection in this setting is persistently difficult because of extreme class imbalance [13].

# 2.2 Graph neural networks

GNNs generalise neural networks to graph-structured data by aggregating information from neighbouring nodes [14]. Graph convolutional networks [15], graph attention networks [16] and GraphSAGE [17] span the main aggregation strategies. For heterogeneous graphs, relational graph convolutional networks [18] and heterogeneous graph transformers [19] introduce type-specific transformations. Two limitations are directly relevant. Deep message passing causes over-smoothing, in which node representations converge [20]. And GNN advantages depend heavily on homophily: where connected nodes do not share labels, standard aggregation can underperform graph-agnostic models [21].

Evaluation methodology for comparing learning algorithms is itself mature. Dietterich [22] showed that naive resampled t-tests over cross-validation folds have inflated Type I error because training sets overlap. Nadeau and Bengio [23] derived the corrected resampled test used here, and Demšar [24] set out the standard framework for comparing classifiers across datasets. These cautions are applied less consistently in domain journals than in machine learning venues.

# 2.3 Graph learning in supply chains

Wasi et al. [2] introduced SupplyGraph, the first benchmark designed for GNN-based supply chain planning. Kosasih and Brintrup [1] applied GNNs to hidden link prediction in supply networks, and Wu et al. [3] proposed heterogeneous GNNs for fraud detection in supply chain finance. Wasi et al. [25] benchmarked homogeneous and heterogeneous models across six tasks, reporting improvements over traditional methods.

Among the studies reviewed for this work, we found few that simultaneously isolate the contribution of graph structure at fixed architecture, enforce temporal feature-label separation, and test whether the specific relations outperform a degree-preserving randomisation. The present study supplies those controls.

## 3. Data and graph construction

Three datasets of contrasting scale and structure are used (Table 1). The DataCo Smart Supply Chain dataset [26] contains 180,519 e-commerce order records spanning January 2015 to January 2018. SupplyGraph [2] provides 221 days of operational data for a fast-moving consumer goods producer with four relation types. Olist [27] is a Brazilian e-commerce dataset with approximately 99,000 delivered orders, real order lifecycle timestamps and an explicit seller entity, which DataCo lacks.

Table 1
Dataset characteristics, graph statistics and evaluation windows. Edge counts are directed and include reverse relations, which are added for every directed relation so that message passing is bidirectional.

| Property | DataCo | SupplyGraph | Olist |
|---|---|---|---|
| Original format | Tabular (180,519 orders) | Graph-structured | Tabular (~ 99,000 orders) |
| Node types used | Product, customer, category | Product | Seller, product, category |
| Nodes | 54 / 9,866 / 24 (total 9,944) | 40 | 776 / 1,156 / 73 (total 2,005) |
| Relation types | 3 semantic + 2 reverse | 4 | 4 semantic + 2 reverse |
| Directed edges (incl. reverses) | 107,460 | 2,504 | 4,830 |
| of which co-purchase | 882 (441 pairs, threshold 10) | - | 74 (37 pairs, threshold 2) |
| Target entities | 54 products | 40 products | 776 sellers / 1,156 products |
| Feature window | Jan 2015 - Jun 2016 | Days 1-191 | to Dec 2017 |
| Label window | Jul 2016 - Aug 2017 | Days 192–221 | Jan - Jul 2018 |
| Label definition | Late-delivery rate above median | See Section 3 | Late-delivery rate above median |
| Outcome source | Derived risk flag | Operational signals | Actual vs estimated delivery date |

# 3.1 Data auditing

Three properties of DataCo materially affect any study using it. First, the supplied late-delivery flag agrees with the rule *actual shipping days exceed scheduled shipping days* for 97.55 per cent of records, so shipping-duration variables are near-deterministic proxies for the label and are excluded from all model inputs. Second, monthly order volume is not stable (Fig. 1): it holds near 5,200 orders per month through August 2017 and then falls to roughly 2,100, with a simultaneous spike and collapse in mean order value. We treat this as truncated data and terminate the analysis window before it. Third, the maximum absolute correlation between individual numeric features and the late-delivery flag is 0.401 (Fig. 2), not below 0.3 as is sometimes asserted in work using this dataset. For SupplyGraph, the node index file lists one product identifier twice; naive loading silently drops one product, so the usable node count is 40, not 41.

# 3.2 Temporal protocol

For each dataset a feature window and a strictly later label window are defined. Node features and all graph edges are computed from the feature window only; labels come from the label window only. An assertion verifies that the last feature-window timestamp precedes the first label-window timestamp.

For DataCo the feature window is January 2015 to June 2016 and the label window July 2016 to August 2017, leaving 54 products with a 28/26 class split. For Olist, features come from orders placed before January 2018 and labels from January to July 2018, giving 776 sellers and 1,156 products. For SupplyGraph the first 191 days form the history window and the final 30 days the label window.

# 3.2.1 Task definitions

Nine prediction tasks are evaluated. On DataCo: **T1**, future product-level late-delivery risk, where a product is positive if its late-delivery rate over the label window exceeds the cross-product median of that window — this is entity-level risk, not per-order prediction, and the distinction matters for interpretation; and **T2**, future customer-level fraud, where a customer is positive if any order placed in the label window is flagged as suspected fraud. T1 is additionally evaluated under two protocol variants

used as sensitivity analyses: a **narrow window** (features to December 2016, labels from January 2017, minimum five orders per window) and a **static protocol** that relaxes temporal separation while retaining the exclusion of outcome-derived features.

On SupplyGraph, with the first 191 days as history and the final 30 as the label window: **T3**, demand forecasting, regressing mean daily sales volume over the label window; **T4**, factory-issue detection, where a product is positive if its proportion of days with a non-zero factory issue in the label window exceeds the cross-product median; and **T5**, production-anomaly detection, where a product is positive if more than half its observed production days in the label window are zero.

On Olist: **O1**, seller-level late-delivery risk, and **O2**, product-level late-delivery risk, each defined by the entity's late-delivery rate over the label window exceeding the cross-entity median. For Olist, late delivery is an observed event — the delivered date exceeding the estimated delivery date — rather than a supplied flag.

## 3.3 Graph schema

For DataCo we construct product, customer and category nodes joined by *belongs_to*, *co_purchased* and *purchased* relations plus reverses. A co-purchase edge links two products appearing in the same order, retained when the co-occurrence count reaches a threshold varied between 5 and 50 in a sensitivity analysis. For Olist we use seller, product and category nodes joined by *sells*, *belongs_to*, product co-purchase and seller co-order relations plus reverses; customers are omitted because Olist issues a customer identifier per order, so a customer–product graph carries almost no repeat structure. For SupplyGraph we retain all four supplied relations as distinct types rather than merging them into a single edge set.

Node features are aggregate commercial descriptors computed over the feature window, with categorical attributes one-hot encoded rather than integer-coded, since integer coding imposes a spurious ordinal relation. No feature derived from the outcome, or from information unavailable at the prediction point, is retained; an assertion enforces this against an explicit exclusion list.

# 4. Method

## 4.1 Models

Three heterogeneous architectures are evaluated: HeteroGraphConv, using the relation-specific operator of Morris et al. [28]; HeteroGAT; and HeteroSAGE. Each has type-specific input projections, relation-specific convolutions aggregated by summation, batch normalisation, dropout and a two-layer classification head. Depth is limited to two layers to mitigate over-smoothing. All figures were generated using Python with Matplotlib and Seaborn. Models are implemented in PyTorch Geometric [29]. Eleven conventional baselines are run using scikit-learn [30], XGBoost [8] and LightGBM [9]: logistic or linear regression, ridge regression at two penalties, decision tree, random forest, k-nearest neighbours, support vector machine, multilayer perceptron, gradient boosting, XGBoost and LightGBM. Trivial baselines (majority, all-positive and a random scorer averaged over 50 draws per fold) establish the chance floor, and historical lag-score baselines are added for the temporal SupplyGraph tasks.

## 4.2 Protocol

Each task is evaluated over five seeds by five stratified folds. Within each fold the training portion is split again to give a validation set. Feature scaling is fitted on training rows only, and test folds are never used for model selection. Hyperparameters are selected from an eight-point grid using validation folds of a designated search seed and then fixed for multi-seed evaluation; this is not fully nested cross-validation and is not described as such. Conventional baselines are fitted on the training portion only, matching the data available to the GNN.

Epoch selection and hyperparameter choice both use validation AUC, matching the metric reported. Selecting on a thresholded metric while reporting a ranking metric would give the graph models no opportunity to optimise what they are judged on, which is indefensible when the analysis concerns a possible graph deficit.

Reporting the best-performing baseline selected from a results table is selection on the test folds. We instead select, within each fold, the conventional model with the highest validation score and record its test performance under a single name. The comparator varies across folds and is never chosen using test data.

## 4.3 Structure ablations

To separate message passing from network depth, five variants are compared at fixed architecture: the full graph; **self-loops only**, in which every node aggregates from itself; **rewired**, a degree-preserving randomisation obtained by permuting the destination array, which preserves both degree sequences exactly; an **approximately capacity-matched feature-only network** with the same depth, normalisation and dropout but dense layers in place of convolutions, automatically sized so its parameter count approximates the GNN's; and **no edges**, which builds zero convolution layers and is therefore confounded with depth. Parameter counts are reported rather than assumed equal.

## 4.4 Statistical analysis

The 25 runs per model are not independent: folds within a seed share most of their training data. We use the Nadeau and Bengio [23] corrected resampled t-test with Holm [31] adjustment within each family of comparisons. Uncertainty is expressed as 95 per cent confidence intervals from a cluster bootstrap resampling seeds; with five seed-level clusters these are descriptive rather than precise inferential bounds. Effect sizes are reported throughout. The correction is conservative by construction, and applying it changed conclusions: 25 of the 45 model comparisons significant under an uncorrected fold-level rank test are not significant under it.

For classification we report AUC and PR-AUC. F1 is not used for ranking, because on a balanced task a constant predictor scores approximately 0.667. For regression we report MAE, since R-squared over roughly eight held-out products is dominated by between-product variance rather than forecasting skill.

## 4.5 Use of generative artificial intelligence

A large language model (Claude, Anthropic) was used as a development and drafting aid in preparing this study. Specifically, it assisted with implementing the analysis pipeline, including the graph construction, evaluation harness, statistical procedures and plotting code; with identifying methodological defects in earlier versions of the pipeline, several of which are documented in Section 6.3; and with drafting and editing manuscript text.

The authors defined the research questions, selected the datasets and tasks, specified the experimental protocol, executed all computational experiments, and reviewed and verified every reported result. All numerical values, tables and figures in this manuscript were checked multiple times against the executed analysis outputs by the authors, and the analysis was independently re-examined across several revisions. The authors take full responsibility for the content, the accuracy of the reported results and the conclusions drawn from them.

The complete analysis code and per-run results are provided in the reproducibility package so that every reported value can be independently regenerated and verified.

# 5. Results

## 5.1 GNNs versus the validation-selected comparator

Table 2 addresses RQ1. On no task does any GNN achieve a difference from the comparator that survives correction.

Table 2
GNN performance against the validation-selected conventional comparator. Differences are GNN minus comparator. All p-values are Nadeau-Bengio corrected and Holm-adjusted within task. n = 25 paired runs throughout. No comparison reaches significance.

| Task | Metric | Comparator (95% CI) | Best GNN | Difference | d | p (Holm) |
|---|---|---|---|---|---|---|
| DataCo T1 late delivery | AUC | 0.779 (0.722–0.815) | HeteroGraphConv 0.702 | -0.078 | -0.44 | 0.45 |
| DataCo T1 late delivery | PR-AUC | 0.776 (0.715–0.825) | HeteroSAGE 0.704 | -0.072 | -0.57 | 0.60 |
| DataCo T1 (narrow window) | AUC | 0.596 | HeteroGraphConv 0.497 | -0.099 | - | n.s. |
| DataCo T1 (static protocol) | AUC | 0.630 (0.615–0.648) | HeteroSAGE 0.626 | -0.004 | -0.04 | 1.00 |
| DataCo T2 fraud | PR-AUC | 0.055 (0.054–0.056) | HeteroGraphConv 0.054 | -0.001 | -0.09 | 1.00 |
| SupplyGraph T3 demand | MAE | 169.1 (141.8-200.6) | HeteroSAGE 178.3 | + 9.2 (worse) | -0.08 | 1.00 |
| SupplyGraph T4 factory issue | AUC | 0.980 (0.965–0.993) | HeteroSAGE 0.950 | -0.030 | -0.36 | 0.86 |
| SupplyGraph T5 anomaly | AUC | 0.946 (0.930–0.963) | HeteroSAGE 0.947 | + 0.001 | + 0.02 | 0.97 |
| Olist O1 seller late delivery | AUC | 0.629 (0.626–0.631) | HeteroSAGE 0.608 | -0.021 | -0.40 | 0.46 |
| Olist O2 product late delivery | AUC | 0.579 (0.566–0.590) | HeteroGAT 0.611 | + 0.032 | + 0.82 | 0.20 |
| Olist O2 product late delivery | PR-AUC | 0.613 (0.599–0.626) | HeteroSAGE 0.627 | + 0.015 | + 0.42 | 1.00 |

Note: for SupplyGraph T4 and T5 the historical lag-score baseline reaches AUC 0.995 and 0.991 respectively, above every learned model.

On DataCo Task 1 the comparator reaches AUC 0.779 (95 per cent CI 0.722–0.815). HeteroGraphConv is closest at 0.702, a difference of − 0.078 ($d$ = − 0.44, Holm-adjusted $p$ = 0.45). The composition of the comparator is itself a finding: no conventional algorithm is selected in more than a third of folds, and the validation-selected comparator reaches 0.779 where the retrospectively best single model reaches 0.795. That gap of 0.016 AUC is the cost of selecting a baseline without access to the test outcome.

On Olist seller-level prediction the comparator reaches 0.629 and the best GNN 0.608 ($d$ = − 0.40, $p$ = 0.46). On Olist product-level prediction the ordering reverses: HeteroGAT reaches 0.611 against the comparator's 0.579, a difference of + 0.032 ($d$ = 0.82, $p$ = 0.20). On PR-AUC the same comparison narrows to between + 0.001 and + 0.015, so the advantage is metric-dependent.

On the SupplyGraph tasks, conventional models and a historical lag-score baseline dominate. For production anomaly detection the lag-score baseline achieves AUC 0.991, higher than every learned model, although HeteroSAGE at 0.947 is indistinguishable from the comparator (+ 0.001, $p$ = 0.97). For demand forecasting the comparator achieves MAE 169.1 against HeteroSAGE's 178.3; HeteroGAT degrades to 453.9 ($d$ = − 1.17). Fraud detection is a null for every method: with a 5.15 per cent base rate the comparator reaches PR-AUC 0.055 and the best GNN 0.054, and nothing exceeds the chance floor. We report this as evidence that future fraud was not separable under this feature set, granularity and protocol, not as a GNN failure.

## 5.2 Does message passing help?

Table 3 and Fig. 3 address RQ2 with the architecture held fixed. The datasets separate sharply.

Table 3
Structure ablations. All variants use HeteroSAGE except the dense control. Differences are full graph minus control, computed before rounding; positive values favour the real graph. Holm adjustment is applied within each dataset across the three tested controls.

| Dataset | Full | Self-loops | Rewired | Dense control | No edges |
|---|---|---|---|---|---|
| DataCo (AUC) | 0.696 | 0.780 | 0.701 | 0.779 | 0.749 |
| Difference vs full | - | -0.083 (d -0.65, p 0.72) | -0.005 (d -0.03, p 1.00) | - | -0.053 (d -0.37, p 1.00) |
| Olist O1 sellers (AUC) | 0.608 | 0.619 | 0.601 | 0.626 | 0.615 |
| Difference vs full | - | -0.012 (d -0.24, p 1.00) | + 0.006 (d + 0.15, p 1.00) | - | -0.007 (d -0.13, p 1.00) |
| Olist O2 products (AUC) | **0.607** | 0.562 | 0.552 | 0.564 | 0.571 |
| Difference vs full | - | **+ 0.045 (d 1.52, p 0.019)** | **+ 0.056 (d 2.18, p 0.001)** | - | **+ 0.036 (d 1.18, p 0.038)** |
| Parameters (DataCo) | 23,490 | 6,850 | 23,490 | 21,946 | 2,306 |
| Parameters (Olist O1) | 7,538 | 2,258 | 7,538 | 7,310 | 1,010 |
| Parameters (Olist O2) | 106,946 | 24,386 | 106,946 | 108,602 | 7,106 |

Across the three tasks, differences from the depth-matched self-loop control range from − 0.083 to + 0.045 AUC. All p-values are Nadeau-Bengio corrected and then Holm-adjusted within dataset across the three tested controls. An inverse-variance pooled summary of these three differences is reported in the supplementary reproducibility package; we do not report it here because pooling assumes a common effect across three heterogeneous tasks.

On **Olist products** the real graph outperforms every control, and all three comparisons survive Holm adjustment within the dataset. Against the degree-preserving rewiring the difference is + 0.056 AUC ($d$ = 2.18, adjusted $p$ = 0.001); against the depth-matched self-loop control, + 0.045 ($d$ = 1.52, $p$ = 0.019); against the edge-free control, + 0.036 ($d$ = 1.18, $p$ = 0.038). The rewiring comparison is the most informative, because it holds degree sequence and edge count fixed and randomises only which nodes are connected. Its result means the specific relations carry information, not merely that some structure is present.

On **DataCo** the full graph reaches 0.696 while self-loops reach 0.780 and the rewired graph 0.701; the real topology performs no better than a randomisation of itself (− 0.005, $p$ = 1.00). On **Olist sellers** all variants fall within 0.02 of one another. Across the three tasks, differences from the self-loop control range from − 0.083 to + 0.045 AUC. We report these per task rather than pooling them, since the tasks differ in dataset, entity type and sample size.

The self-loop control necessarily has fewer parameters than the full model, because removing the heterogeneous edge types also removes their relation-specific transformation matrices within the heterogeneous convolution. The dense control exists precisely to close that loophole: it matches the full model's parameter budget and reaches almost the same AUC on Olist products (0.564 against 0.562 for self-loops), so the deficit of both controls relative to the full graph cannot be attributed to capacity. Parameter counts confirm this: 23,490 against 21,946 on DataCo, 7,538 against 7,310 on Olist sellers and 106,946 against 108,602 on Olist products, differences of 7, 3 and 2 per cent in no consistent direction.

## 5.3 Topology diagnostics

Table 4 addresses RQ3. For each relation we project the graph onto the target nodes — bipartite relations through their shared neighbour, which is what two-layer message passing exploits — and measure density, connectivity, edge homophily against its chance value, and how well the neighbour positive-rate alone ranks the label.

Table 4
Topology diagnostics for the union projection of all relations onto the target nodes

| Dataset | Edges | Density | Mean degree | Isolated | Components | Homophily | Chance | Excess | Neighbour-only AUC |
|---|---|---|---|---|---|---|---|---|---|
| DataCo T1 (product) | 2,812 | 0.982 | 52.07 | 0 | 1 | 0.489 | 0.501 | −0.011 | 0.047 |
| SupplyGraph T5 (product) | 1,368 | 0.877 | 34.20 | 0 | 1 | 0.450 | 0.501 | −0.051 | 0.045 |
| Olist O1 (seller) | 100 | 0.0002 | 0.13 | 775 | 732 | 0.500 | 0.500 | 0.000 | 0.466 |
| Olist O2 (product) | 75,370 | 0.056 | 65.20 | 8 | 17 | 0.508 | 0.500 | + 0.008 | 0.581 |

Note: the low neighbour-only AUC values for DataCo and SupplyGraph are a consequence of near-completeness, not evidence of heterophily. See Section 5.3. Only the Olist product projection exceeds chance, and it is the only dataset on which the structure ablations separate.

Two distinct degeneracies appear. The DataCo and SupplyGraph projections are **near-complete**, at densities of 0.982 and 0.877 with mean degrees of 52 out of 54 nodes and 34 out of 40. When almost every node neighbours almost every other, aggregated messages are nearly identical across nodes and cannot discriminate between them. The Olist seller projection is the opposite: density 0.0002, mean degree 0.13, and 775 of 776 sellers isolated across 732 components. Only the Olist product projection is moderately structured — density 0.056, 17 components, homophily excess + 0.008 and neighbour-only AUC 0.581 — and it is the only projection exceeding chance and the only dataset where message passing significantly outperforms its controls.

Across the four dataset-level observations, neighbourhood informativeness and message-passing benefit are positively associated (Spearman $\rho = 0.775$, $p = 0.225$; Fig. 4). Measured on raw neighbour-only AUC the association weakens to $\rho = 0.400$. With four points neither is inferential. The regime classification carries the argument; the correlation coefficient does not.

Table 5
Selected relation-level diagnostics. Unlike the union projections in Table 4, several individual relations carry above-chance neighbourhood signal, yet the unions in which they sit do not. Heterogeneous message passing does not operate on a union, so relational signal can be diluted by dense or uninformative relations combined with it.

| Dataset | Relation | Edges | Density | Homophily excess | Neighbour-only AUC |
|---|---|---|---|---|---|
| DataCo | product-co_purchased-product | 882 | 0.308 | -0.031 | 0.339 |
| DataCo | customer-purchased-product | 2,808 | 0.981 | -0.011 | 0.047 |
| Olist O2 | product-co_purchased-product | 74 | 0.0001 | + 0.176 | 0.592 |
| Olist O2 | seller-sells-product | 12,388 | 0.009 | + 0.034 | **0.622** |
| SupplyGraph | product-subgroup-product | 96 | 0.062 | + 0.082 | **0.658** |
| SupplyGraph | product-storage-product | 1,330 | 0.853 | -0.056 | 0.053 |

Individual relations tell a more nuanced story than the union projection (Table 5). Several exhibit neighbour-only AUC above chance, including the Olist seller–product relation (0.622) and the SupplyGraph product–subgroup relation (0.658), yet the unions in which they sit remain uninformative. Heterogeneous message passing does not operate on a simple union, so a relation carrying signal can be diluted when combined with dense or uninformative relations. The single-relation ablations plotted in Fig. 4 cover only the relations that connect target nodes to one another directly, because a bipartite relation such as seller–sells–product cannot be ablated in isolation without also specifying how its two-hop projection is formed. Two homogeneous relations that do exceed chance were also outside the ablation set: product–co_purchased–product on Olist

products carries only 74 edges across 1,156 nodes (density 0.0001) and product–subgroup–product on SupplyGraph only 96 edges across 40 nodes, so neither supports a single-relation model in which most target nodes have any neighbour at all. Running single-relation ablations for the informative bipartite relations is therefore left to future work, and the association reported here should be read as covering the denser homogeneous relations only.

One artefact should be noted. The near-complete projections yield very low neighbour-only AUC values (0.047 and 0.045), which superficially resemble strong heterophily. In a near-complete graph a positive node's neighbourhood necessarily excludes itself, depressing its neighbour positive-rate. This is a consequence of density, not evidence that supply chain relations are heterophilous.

## 5.4 Robustness

Removing test nodes and all their incident edges during training changes AUC by between − 0.005 and + 0.003 across the Olist model-task pairs (Table 6), so conclusions do not depend on test nodes being present in the training graph; this is not evidence of inductive competence. Under the narrow DataCo window the best conventional model reaches only 0.596 and the best GNN 0.497, and on 4 of 25 folds no conventional model beat a constant predictor on validation. Varying the co-purchase threshold across 5, 10, 20 and 50 changes the retained edge count by an order of magnitude without altering the conclusion. No architecture leads consistently across datasets, which argues against the explanation that a single aggregation mechanism is the limiting factor.

Table 6
Transductive versus inductive evaluation (Olist, AUC). Test nodes and all their incident edges are removed during training and rejoin only for scoring.

| Task | Model | Transductive | Inductive | Change |
|---|---|---|---|---|
| O1 seller | HeteroSAGE | 0.608 | 0.604 | -0.004 |
| O1 seller | HeteroGAT | 0.601 | 0.603 | + 0.003 |
| O2 product | HeteroSAGE | 0.607 | 0.606 | -0.001 |
| O2 product | HeteroGAT | 0.611 | 0.606 | -0.005 |

# 6. Discussion

## 6.1 Interpretation

Four explanations of a null are available: the constructed relations carry little task-relevant information; the edges are informative but noisy; node features already contain what the neighbourhood would add; or the architecture cannot extract the signal.

The controls address these jointly, and the answer differs by dataset. On Olist products the rewiring comparison provides evidence against explanations based solely on degree structure or architectural inability: the real graph beats a degree-preserving randomisation of itself at $d = 2.18$, so the particular arrangement of relations carries signal the tested architecture can extract. On DataCo and Olist sellers the same control shows no separation, which is more consistent with the first explanation than with a topology-specific contribution, although the conservative inference procedure limits what can be concluded from a null comparison.

The defensible claim is not that GNNs are unsuitable for supply chain prediction, nor that relational structure is worthless. It is that heterogeneous relational structure does not automatically provide predictive value beyond historical node features, that whether it does is conditional on properties of the constructed graph, and that those properties are measurable before any model is trained.

Two results must be held apart. Message passing significantly outperforms its own controls on Olist products. The same graph models do not significantly outperform a validation-selected conventional baseline on that task or any other. Relational information contributes where the graph carries it, and the contribution was not large enough to change which model an analyst should deploy.

Constructing a graph from tabular transactions involves a threshold choice trading between two degeneracies. Permissive construction on a small entity set produces near-complete projections, as with 54 DataCo products linked through 9,866 shared customers. Restrictive construction on sparse co-occurrence produces near-empty graphs, as with Olist sellers, where most orders contain a single item. Neither extreme leaves usable neighbourhood signal.

## 6.2 Practical implications

Three implications follow for practitioners considering graph analytics in logistics functions.

**Diagnose the graph before investing in the model.** Graph learning projects carry real costs: specialist skills, graph infrastructure, longer training cycles and models harder to explain to operational stakeholders. The diagnostics in Table 4 take seconds to compute from data an organisation already holds and provide an inexpensive way to characterise whether a constructed graph contains measurable neighbourhood signal.

**Entity granularity drives topology more than data volume does.** DataCo contains 180,519 orders, an order of magnitude more transactional data than the Olist product task, yet its product projection is degenerate because 54 products linked through thousands of shared customers produce a near-clique. The binding constraint is the ratio between the number of target entities and the breadth of the attributes connecting them, not the number of rows. Practitioners choosing a prediction granularity are implicitly choosing a topology.

**Simple temporal baselines are strong and frequently omitted.** On the production anomaly task a historical lag-score baseline outperformed every learned model. Any business case for a learned model should quantify improvement over such a baseline, not over a naive constant.

## 6.3 Relation to prior reports

Reported gains for GNNs on supply chain tasks may in some cases reflect evaluation design as well as relational learning. We make no claim about any specific published study; the following mechanisms are ones we observed and corrected in our own earlier iterations. Aggregating features and labels over the same period makes a feature a near-deterministic function of the label. Selecting the best training epoch on the test fold inflates reported performance. Comparing against baselines with a different feature set confounds the comparison. Selecting the best baseline retrospectively from test results is winner's-curse selection. A fifth mechanism operates in the opposite direction: selecting graph-model epochs on a thresholded metric while reporting a ranking metric understates graph performance, and correcting it changed several conclusions in this study.

## 6.4 Limitations

The DataCo product-level task retains only 54 nodes, and no temporal split of this dataset yields materially more, because only 54 of 118 products appear in both an early and a later period. Olist carries the statistical weight. The corrected resampled test is conservative, so several differences with moderate effect sizes are reported as no difference; this reflects the design's resolution, not evidence of equivalence. The evaluation is transductive except where stated. The positive Olist product result rests on one dataset and one metric: it is visible on AUC and absent on PR-AUC, and we have not established why the two diverge. Replication on further moderately structured supply chain graphs would be required before treating the effect as general. SupplyGraph results rest on 40 nodes and are a small-scale external check, not primary evidence.

# 7. Conclusion

We evaluated heterogeneous graph learning for supply chain prediction under a protocol that separates feature and label windows, gives conventional baselines and GNNs identical features and training data, selects the comparator on validation

rather than test performance, aligns the model-selection metric with the reported metric, and corrects inference for cross-validation dependence.

Across three datasets and nine tasks, no GNN achieved a statistically distinguishable advantage over a validation-selected conventional comparator. Controlled ablations holding architecture constant show that the contribution of message passing is conditional. On the one dataset whose target-node projection carries above-chance neighbourhood signal, the real graph significantly outperforms a degree-preserving randomisation of itself, a depth-matched self-loop control and an edge-free control, with effect sizes between 1.18 and 2.18. On two near-complete projections and one near-empty projection no such benefit is detectable.

For the datasets evaluated here, graph construction appears to be a more important limitation than the choice among the tested aggregation architectures. Before adopting graph learning on a constructed graph, measure whether the topology carries label-relevant neighbourhood signal. Where it does not, a well-tuned tabular model with the same node features was competitive with, and often outperformed, the tested GNNs while requiring less graph-specific infrastructure.

Future work should test whether relations derived from genuine operational links — shared carriers, physical routes, contractual supplier relationships — avoid the degeneracies observed in relations inferred from co-occurrence, and should establish why the Olist product advantage is visible on AUC and not on PR-AUC.

### Declarations

## Competing interests

The authors declare no competing financial or non-financial interests directly or indirectly related to the work submitted for publication.

## Ethics approval

Not applicable. The study uses publicly available benchmark datasets and involves no human or animal participants.

## Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

## Author Contribution

A.B.S., G.S.K., P.C., M.S.K., M.A.M., M.Z.H.J., M.W., and M.I.F. contributed to the study conception and design. Material preparation, data curation, and software implementation were performed by A.B.S., G.S.K., P.C., and M.Z.H.J. Formal analysis and investigation were conducted by A.B.S., M.S.K., M.A.M., and M.W. The first draft of the manuscript was written by A.B.S., and all authors commented on and critically revised previous versions of the text. All authors read and approved the final manuscript.

## Data Availability

The DataCo Smart Supply Chain dataset is available from Mendeley Data (https://doi.org/10.17632/8gx2fvg2k6.5). The SupplyGraph benchmark and the Olist Brazilian e-commerce dataset are publicly available. All analysis code, per-run results, topology diagnostics and SHA256 checksums of every source file are provided in the reproducibility package, which will be released publicly on acceptance.

## References


1. Kosasih, E.E., Brintrup, A.: A machine learning approach for predicting hidden links in supply chain with graph neural networks. Int. J. Prod. Res. 60(17), 5380–5393 (2022). https://doi.org/10.1080/00207543.2021.1956697
2. Wasi, A.T., Islam, M.S., Akib, A.R.: SupplyGraph: a benchmark dataset for supply chain planning using graph neural networks. In: AAAI 2024 Workshop on Graphs and More Complex Structures for Learning and Reasoning (2024). https://doi.org/10.48550/arXiv.2401.15299
3. Wu, B., Chao, K.M., Li, Y.: Heterogeneous graph neural networks for fraud detection and explanation in supply chain finance. Inf. Syst. 121, 102335 (2024). https://doi.org/10.1016/j.is.2023.102335
4. Kaufman, S., Rosset, S., Perlich, C., Stitelman, O.: Leakage in data mining: formulation, detection, and avoidance. ACM Trans. Knowl. Discov. Data 6(4), 15 (2012). https://doi.org/10.1145/2382577.2382579
5. Carbonneau, R., Laframboise, K., Vahidov, R.: Application of machine learning techniques for supply chain demand forecasting. Eur. J. Oper. Res. 184(3), 1140–1154 (2008). https://doi.org/10.1016/j.ejor.2006.12.004
6. Feizabadi, J.: Machine learning demand forecasting and supply chain performance. Int. J. Logist. Res. Appl. 25(2), 119–142 (2022). https://doi.org/10.1080/13675567.2020.1803246
7. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
8. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, New York (2016). https://doi.org/10.1145/2939672.2939785
9. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: a highly efficient gradient boosting decision tree. Adv. Neural Inf. Process. Syst. 30, 3146–3154 (2017). https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html
10. Ni, D., Xiao, Z., Lim, M.K.: A systematic review of the research trends of machine learning in supply chain management. Int. J. Mach. Learn. Cybern. 11, 1463–1482 (2020). https://doi.org/10.1007/s13042-019-01050-0
11. Shwartz-Ziv, R., Armon, A.: Tabular data: deep learning is not all you need. Inf. Fusion 81, 84–90 (2022). https://doi.org/10.1016/j.inffus.2021.11.011
12. Hosseinnia Shavaki, F., Ebrahimi Ghahnavieh, A.: Applications of deep learning into supply chain management: a systematic literature review and a framework for future research. Artif. Intell. Rev. 56, 4447–4489 (2023). https://doi.org/10.1007/s10462-022-10289-z
13. Pourhabibi, T., Ong, K.L., Kam, B.H., Boo, Y.L.: Fraud detection: a systematic literature review of graph-based anomaly detection approaches. Decis. Support Syst. 133, 113303 (2020). https://doi.org/10.1016/j.dss.2020.113303
14. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. 32(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386
15. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017). https://doi.org/10.48550/arXiv.1609.02907
16. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018). https://doi.org/10.48550/arXiv.1710.10903
17. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. 30, 1024–1034 (2017). https://doi.org/10.48550/arXiv.1706.02216
18. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: The Semantic Web: ESWC 2018, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
19. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: Proceedings of The Web Conference 2020, pp. 2704–2710. ACM, New York (2020). https://doi.org/10.1145/3366423.3380027

20. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3538–3545. AAAI Press, Palo Alto (2018). https://doi.org/10.1609/aaai.v32i1.11604
21. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: current limitations and effective designs. Adv. Neural Inf. Process. Syst. 33, 7793–7804 (2020). https://doi.org/10.48550/arXiv.2006.11468
22. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10(7), 1895–1923 (1998). https://doi.org/10.1162/089976698300017197
23. Nadeau, C., Bengio, Y.: Inference for the generalization error. Mach. Learn. 52(3), 239–281 (2003). https://doi.org/10.1023/A:1024068626366
24. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (2006). https://www.jmlr.org/papers/v7/demsar06a.html
25. Wasi, A.T., Islam, M.S., Akib, A.R., Bappy, M.M.: Graph neural networks in supply chain analytics and optimization: concepts, perspectives, dataset and benchmarks. arXiv:2411.08550 (2024). https://doi.org/10.48550/arXiv.2411.08550
26. Constante, F., Silva, F., Pereira, A.: DataCo Smart Supply Chain for Big Data Analysis. Mendeley Data, V5 (2019). https://doi.org/10.17632/8gx2fvg2k6.5
27. Olist, Sionek, A.: Brazilian E-Commerce Public Dataset by Olist. Kaggle (2018). https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce
28. Morris, C., Ritzert, M., Fey, M., Hamilton, W.L., Lenssen, J.E., Rattan, G., Grohe, M.: Weisfeiler and Leman go neural: higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4602–4609. AAAI Press, Palo Alto (2019). https://doi.org/10.1609/aaai.v33i01.33014602
29. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019). https://doi.org/10.48550/arXiv.1903.02428
30. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011). https://www.jmlr.org/papers/v12/pedregosa11a.html
31. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. 6(2), 65–70 (1979). https://doi.org/10.2307/4615733

## Figures

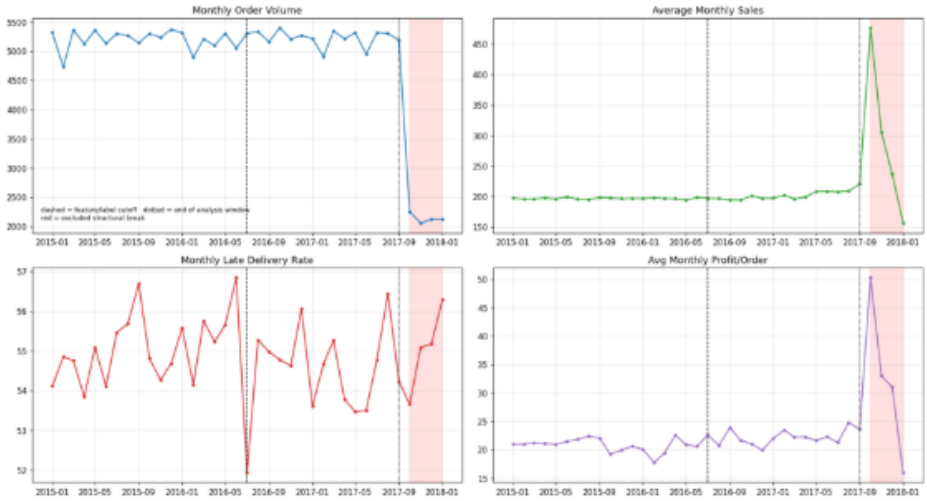


**Figure 1**

DataCo temporal trends. Monthly order volume, mean order value, late-delivery rate and mean profit per order. The dashed line marks the feature/label cutoff, the dotted line the end of the analysis window, and the shaded region the excluded structural break from September 2017

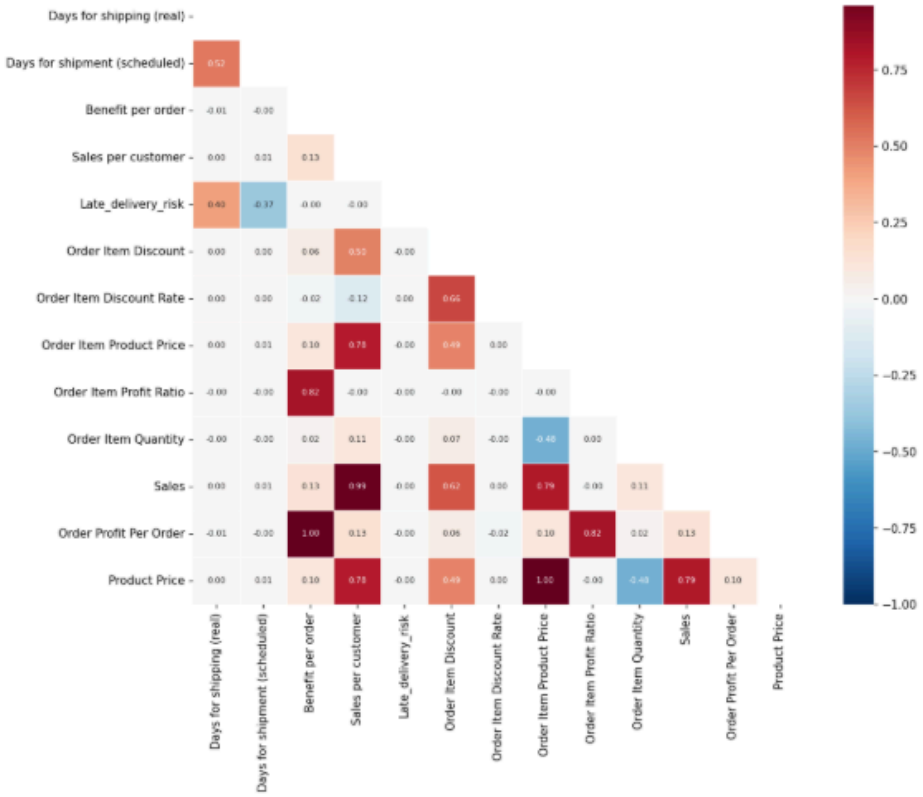


**Figure 2**

DataCo correlation matrix for key numeric features, annotated. The maximum absolute correlation with the late-delivery flag is 0.401, against actual shipping days

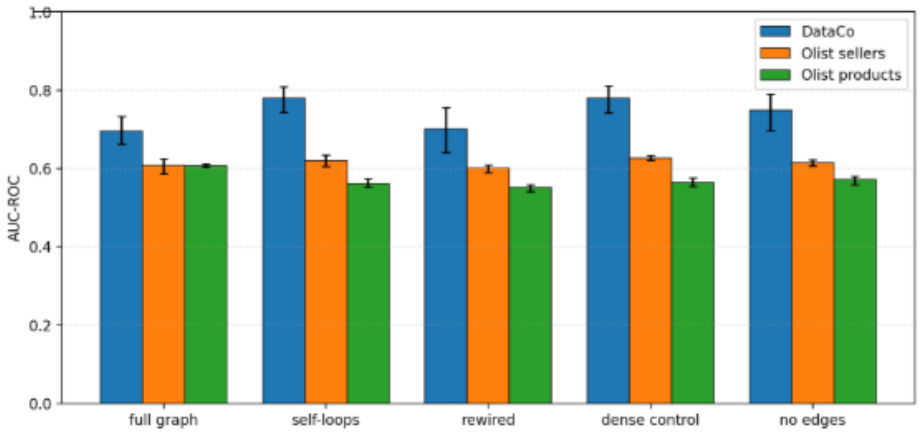


**Figure 3**

Structure ablations by dataset. The architecture is held constant and only the edge content varies: the real graph, self-loops only, a degree-preserving randomisation, an approximately capacity-matched dense control, and no edges. Error bars are 95% cluster-bootstrap intervals over five seed-level clusters. On Olist products the real graph exceeds all three tested controls with Holm-adjusted significance; on DataCo and Olist sellers no comparison separates

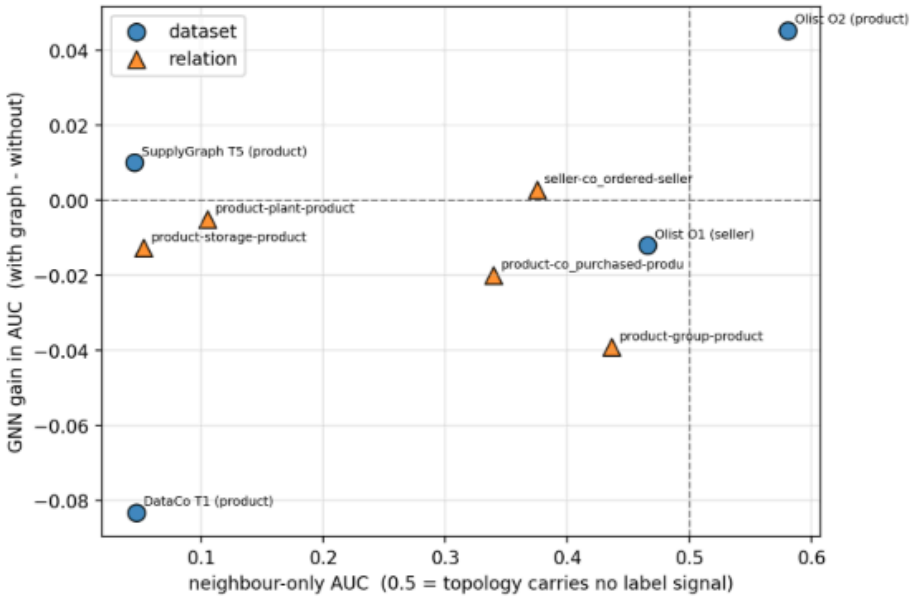


**Figure 4**

Neighbourhood informativeness against observed message-passing benefit. The horizontal axis is the neighbour-only AUC of the target-node projection, where 0.5 indicates that the topology carries no label signal. The vertical axis is the AUC difference between the full graph and the depth-matched self-loop control. Circles are the four dataset-level union projections. Triangles are the five relations for which a matched single-relation ablation was run: DataCo product-co-purchase (0.339), Olist seller-co-order (0.375), SupplyGraph plant (0.105), storage (0.053) and group (0.436). These five are **not** the relations listed in Table 5, which reports the most and least informative relations per dataset and includes three that exceed chance. Every relation plotted here falls below chance on the horizontal axis, so no association is estimated among them. The dataset-level association is positive (Spearman rho = 0.775 on excess signal, p = 0.225; rho = 0.400 on raw neighbour-only AUC, p = 0.600). The only observation with a clear positive benefit is also the only projection exceeding chance. With four dataset-level points the relationship is illustrative rather than inferential; the regime classification in Table 4 carries the argument