



Article

Is There a Best Hypergraph Neural Network? A Significance-Aware Recomputation and Statistical Audit of DHG-Bench

Valeriya V. Tynchenko ^{1,2} , Sergei O. Kurashkin ² , Aleksei S. Borodulin ² , Tee Connie ³ , Ahmad Hammoud ⁴ and Vadim S. Tynchenko ^{2,*}

¹ Department of Computer Science, Institute of Space and Information Technologies, Siberian Federal University, 660041 Krasnoyarsk, Russia; 051301@mail.ru

² Artificial Intelligence Technology Scientific and Education Center, Bauman Moscow State Technical University, 105005 Moscow, Russia; kurashkin@bmstu.ru (S.O.K.); alexey.borodulin@emtc.ru (A.S.B.)

³ Faculty of Information Science and Technology, Multimedia University, Cyberjaya 63100, Malaysia; tee.connie@mmu.edu.my

⁴ Department of Medical and Technical Information Technology, Bauman Moscow State Technical University, 105005 Moscow, Russia; hammoud@bmstu.ru

* Correspondence: vadimond@mail.ru

Abstract

Deep hypergraph learning is evaluated almost entirely through leaderboards that rank methods by mean accuracy over a few random seeds, usually without significance testing. Is there a best hypergraph neural network, or does the apparent ordering reflect seed noise? We independently recomputed the node-classification track of DHG-Bench on a single GPU with twenty random seeds (against five upstream) and a different software stack, and applied a four-layer statistical audit to the per-seed accuracies: a reproducibility check, per-dataset paired Wilcoxon tests with Holm correction, an across-datasets Friedman/Iman–Davenport omnibus with Nemenyi and Holm-corrected pairwise tests, and a variance decomposition. Within a single dataset, twenty seeds distinguish most method pairs (74–98%), so the protocol is not underpowered. Across the nine datasets where all 17 methods complete, the omnibus rejects global equality (Kendall’s $W = 0.45$), yet no pair survives Holm correction, and the top methods fall within one critical-difference band. One dataset carries more seed noise than between-method signal and cannot rank methods. The recompute also documents a non-reproducible method, a label-range data fault, and missing per-dataset configurations in the public release. No single method is statistically best across these datasets, so single-leader claims are not supported; we release a reusable significance-aware evaluation protocol.

Keywords: hypergraph neural networks; benchmark evaluation; statistical significance testing; reproducibility; Friedman test; critical-difference diagram; variance decomposition; node classification; leaderboard



Received: 6 June 2026

Revised: 18 July 2026

Accepted: 30 July 2026

Published: 31 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Progress in graph and hypergraph representation learning is reported almost entirely through leaderboards: a benchmark fixes a set of datasets and an evaluation pipeline, each method is run a small number of times, and the methods are ordered by mean

accuracy. The reliability of this practice has been questioned repeatedly. Controlled re-evaluations have shown that apparent rankings of graph neural networks are sensitive to the training and selection protocol, that performance differences often lie within the run-to-run variation, and that simple, well-tuned baselines frequently match far more elaborate models [1–3]. Large standardised benchmarks reduce some of this noise [4], but the broader concern remains: when many methods are compared on a shared suite, the method that tops a leaderboard need not be meaningfully better than its neighbours, and which method appears first can depend on incidental choices [5,6].

A recurring symptom is the near-absence of formal significance testing. Methods are typically reported as the mean $\pm$ standard deviation over a handful of seeds, and conclusions are drawn from overlapping intervals without a test of whether the differences are larger than the seed-to-seed noise. The statistical machinery for this comparison is long established: the Wilcoxon signed-rank test for two methods, the Friedman test with its Iman–Davenport correction for an omnibus comparison across datasets, and post hoc procedures (Nemenyi critical-difference diagrams, or pairwise tests with Holm correction) to locate the differences [7,8]. These tools are standard in the wider machine learning literature but are applied only rarely in graph-learning benchmarks, and the choice of post hoc procedure itself affects which differences are declared significant [9].

Hypergraph neural networks (HNNs), which generalise message passing from pairwise edges to higher-order relations, have inherited both the rapid method growth and the evaluation practices of the graph-learning field. The recent DHG-Bench benchmark [10] is the most systematic attempt to compare them: it places 17 HNNs on a unified pipeline across a common set of datasets and reports node classification, among other tasks, as the mean $\pm$ standard deviation over five seeds. Earlier work such as AllSet [11] had already argued for multi-run evaluation and for including heterophilic datasets. DHG-Bench standardises preprocessing and thereby makes the comparison fairer than earlier ad hoc tables, but it still ranks methods by their means and draws leader conclusions without an accompanying significance analysis. This leaves open the central question of whether the resulting ordering reflects real differences between methods or the noise of a small number of seeds.

This paper addresses that question for the node-classification track of DHG-Bench. We do not propose a new model. Instead, we recompute the track independently and add the statistical layer that the benchmark lacks. The recompute uses twenty random seeds rather than five and a deliberately different software stack, so that both the power of the comparison and its reproducibility can be examined. On top of the recomputed per-seed accuracies we apply a four-layer audit: (i) a reproducibility check against the reported means; (ii) per-dataset paired significance testing across seeds, which is well posed because each seed fixes a common data split; (iii) an across-datasets comparison restricted to the datasets on which all methods complete, using the Friedman/Iman–Davenport omnibus test, Kendall’s coefficient of concordance, a Nemenyi critical-difference diagram, and pairwise Wilcoxon tests with Holm correction; and (iv) a variance decomposition that contrasts the spread between methods with the seed-to-seed noise within methods, per dataset. We report the same machinery as a significance-aware leaderboard that can be reused by the hypergraph-learning community.

The analysis yields a consistent picture. Within a single dataset, given twenty seeds, most pairs of methods are statistically distinguishable, which confirms that the per-seed protocol has adequate power. Across datasets, however, the leaderboard does not survive correction: over the complete block of datasets on which all 17 HNNs return a value, no pair of methods is separable by the Holm-corrected Wilcoxon test, and the top-ranked methods fall inside a single critical-difference band. At least one benchmark dataset cannot rank

methods at all, because its seed-to-seed noise exceeds the typical gap between methods. The recompute additionally exposes a method whose accuracy is not reproducible under a modern stack and a data-pipeline fault on one dataset that prevents almost all methods from training. Together, these results indicate that the heuristic leader claims attached to such benchmarks are not supported once significance is accounted for, while the per-dataset and variance-aware views remain informative.

The contributions of this paper are the following:

- An independent twenty-seed recompute of the DHG-Bench node-classification track under a modern software stack, with per-seed results released in full;
- A four-layer statistical audit (reproducibility, per-dataset paired significance, across-datasets comparison, and variance decomposition) applied to a hypergraph benchmark, with the conclusion that no HNN pair is significantly separable across the complete-block datasets;
- The identification of a benchmark dataset whose seed noise exceeds the between-method gap, and is therefore unable to order methods;
- An analysis relating the hypergraph-structural properties of each dataset to its rankability and to which architecture family leads, from which we distil partial, explicitly-caveated model-selection heuristics;
- A reproducibility audit that documents a non-reproducible method and a data-pipeline fault;
- A significance-aware leaderboard protocol, with code, that the hypergraph-learning community can reuse.

Figure 1 summarises this study: an independent twenty-seed recompute of the DHG-Bench node-classification track feeds four analysis layers, which together decide whether any single method can be called the best.

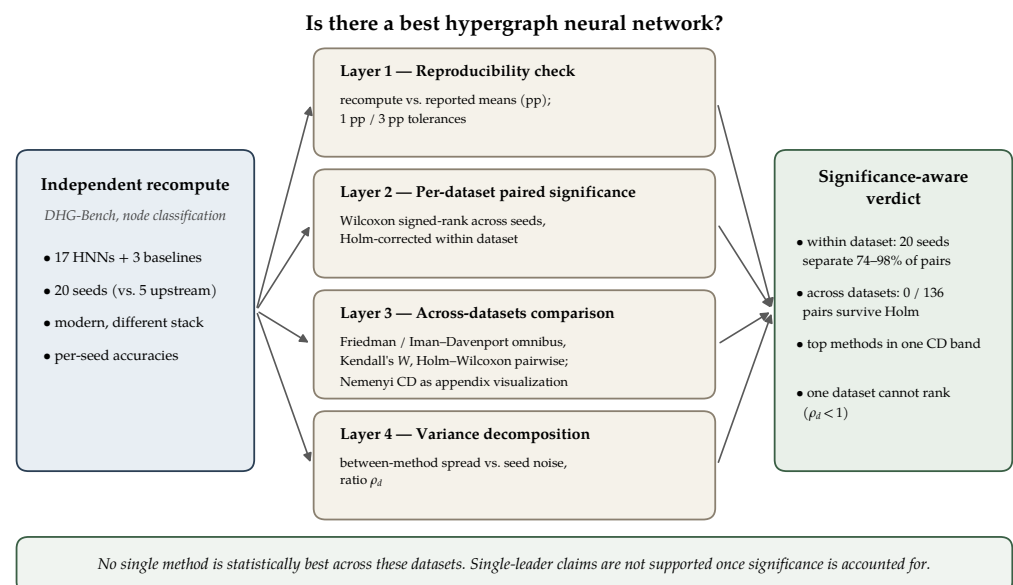


Figure 1. Overview of the significance-aware audit. An independent twenty-seed recompute of the DHG-Bench node-classification track feeds a four-layer statistical analysis (reproducibility, per-dataset paired significance, across-datasets comparison, and variance decomposition). The layers together establish that no single hypergraph neural network is statistically best across the benchmark datasets.

2. Related Work

2.1. Rigour in Graph-Learning Evaluation

The reliability of graph-learning benchmarks has been examined from several angles. Shchur et al. [1] showed that the ranking of graph neural networks on standard node-classification datasets changes substantially with the train/validation split and the hyperparameter budget, and that several architectures become statistically indistinguishable once the evaluation is controlled. Errica et al. [2] reached a parallel conclusion for graph classification, finding that structure-agnostic baselines are competitive with graph neural networks when all models are tuned and evaluated under a common protocol. A recurring theme in the literature is that strong, well-tuned baselines often match more elaborate models: Luo et al. [12] recently showed that classic graph neural networks, given careful hyperparameter tuning, match or exceed graph transformers on most of the eighteen node-classification datasets, attributing the apparent gap to suboptimal baseline configurations. Platonov et al. [13] found that widely used heterophily benchmarks contain data-quality defects, including duplicate nodes that cause train–test leakage, and that conclusions drawn from them are unreliable; Lim et al. [14] likewise argued that the small set of non-homophilous datasets in common use is inadequate for evaluating such methods. Dwivedi et al. [3] introduced a medium-scale benchmark suite with a fixed pipeline to make architectural comparisons reproducible, and the Open Graph Benchmark [4] standardised large-scale datasets and split protocols for the same purpose. At a higher level, the benchmark-lottery analysis of Dehghani et al. [5] argued that the choice of datasets and tasks in a benchmark can determine which method appears to be the best, independently of any real difference in capability; related work has questioned whether graph structure is even necessary on some benchmark datasets [15], and a recent position paper warns that weak benchmarking practices threaten the relevance of graph learning as a field [16]. Our study is in this tradition but differs in instrument: rather than retuning models or redesigning the benchmark, we apply significance testing and a variance decomposition to the per-seed outputs of an existing benchmark, and ask whether its leaderboard ordering is statistically supported. A companion systematic review by the present authors surveys the evaluation-rigour literature across graph and graph-foundation models and proposes a four-axis reporting standard [17]; the present paper is its focused empirical counterpart.

2.2. Statistical Comparison of Learning Algorithms

The statistical tools for comparing classifiers across datasets are well established. Demšar [7] recommended nonparametric procedures (the Wilcoxon signed-rank test [18] for two methods and the Friedman test for several methods over multiple datasets) together with the Nemenyi post hoc test [19] and its critical-difference diagram. García and Herrera [8] extended this framework to all pairwise comparisons with more powerful multiple-comparison corrections, including step-down procedures of the Holm type [20], and later quantified the relative power of these nonparametric tests in detail [21]. The Friedman statistic is commonly used with the less conservative Iman–Davenport correction [22]. Benavoli et al. [9] cautioned that post hoc tests based on mean ranks, such as Nemenyi, can be unstable and low-powered, and recommended pairwise tests with multiplicity control instead, a recommendation we follow by treating the Holm-corrected Wilcoxon tests, rather than the critical-difference diagram, as the primary pairwise analysis. Complementing significance testing, Bouthillier et al. [6] quantified how much of the apparent difference between methods is attributable to sources of variance in the experimental pipeline, and argued that benchmark conclusions should account for this variance explicitly. The same significance- and heterogeneity-aware lens has recently been turned on benchmarks in adjacent fields: a meta-analysis of the SMAC benchmark in cooperative multi-agent re-

inforcement learning found pervasive statistical heterogeneity across studies and broad prediction intervals, and called for standardised, heterogeneity-aware reporting [23]. These tools are standard in the broader machine learning literature but are seldom attached to graph- or hypergraph-learning leaderboards; applying them here is the methodological core of this work.

2.3. Hypergraph Neural Networks and Their Evaluation

Hypergraph neural networks generalise message passing to higher-order relations, and the family has grown quickly [24,25]. Representative designs include the spectral convolution of HGNN [26], the training procedure of HyperGCN for convolution on hypergraphs [27], the hyperedge-neuron model HNHN [28], the unified graph/hypergraph framework UniGNN [29], the equivariant diffusion operator ED-HNN [30], the multiset-function framework of AllSet [11], the tensor-representation model TMPHN [31], and a range of later sheaf- and transformer-based models; the benchmark we audit collects 17 such methods together with non-hypergraph baselines, and recent work has begun to extend the family to heterophilous hypergraphs [32]. AllSet [11] already argued for multi-run evaluation and for including heterophilic datasets, anticipating part of the evaluation-rigour concern. DHG-Bench [10] is, to our knowledge, the most systematic comparison of hypergraph neural networks to date: it standardises preprocessing, splits, and training across node-, edge-, and graph-level tasks, which removes much of the ad hoc variation in earlier tables. It reports node-classification accuracy as the mean and standard deviation over five seeds and orders methods accordingly, but does not attach a significance analysis to the ordering. The present paper supplies that missing layer for the node-classification track and, in doing so, audits the reproducibility of the benchmark's released code and configurations.

3. Materials and Methods

3.1. Object of the Audit

We re-analysed the node-classification track of DHG-Bench [10], a benchmark that evaluates deep hypergraph learning under a common training and evaluation pipeline. The published benchmark was treated as the object of study: its training procedure, data splits, and per-method hyperparameters were left unchanged except for the two minimal, environment-gated patches described in Section 3.2. The track comprises 17 hypergraph neural networks (HNNs): AllSetformer [11], DPHGNN, EDHNN, EHNN, HCHA, HGNN, HJRL, HNHN, HyperGCN, HyperGT, HyperND, LEGCN, PhenomNN, SheafHyperGNN, TFHNN, TMPHN, and UniGCNII. It also includes three non-hypergraph baselines: a feature-only multilayer perceptron (MLP) and the clique-expansion graph models CEGCN and CEGAT. The original references for each method are those cited in the benchmark's method catalogue. The original benchmark reports test accuracy as the mean $\pm$ standard deviation over five random seeds and orders methods by these means; it does not attach a significance analysis to the ordering.

3.2. Recomputation Protocol

We recomputed the node-classification track independently on a single NVIDIA A100 (40 GB). Two protocol choices depart from the upstream evaluation by design. First, we increased the number of random seeds from five to twenty, which raises the statistical power of the per-seed analyses in Section 3.3. Second, we executed the benchmark under a deliberately different software stack (Python 3.12.3 with PyTorch 2.4.0+cu121, against the upstream Python 3.9.21 with PyTorch 2.2.2), so that any large divergence from the re-

ported means is observable as part of the reproducibility audit rather than masked by an identical environment.

Every method was run in the configuration that the public harness exposes by default (the `is_default` setting), so that all methods are compared under a single, common hyperparameter set on each dataset. The public repository ships per-method configuration files containing only two sections, a global default and a dataset-specific section for cora; no dataset-specific section is provided for any of the other twelve node-classification datasets. The harness documentation states that disabling the default setting loads dataset-specific parameters, but for every dataset other than cora, those parameters are absent from the distributed configuration files, so the default set is the only configuration that can be reproduced from the public release. This choice and its consequences for the reproducibility check are revisited in Sections 4.1 and 5.

The upstream harness retains only the mean and standard deviation of each cell. We extended it with an environment-gated hook that writes, for every (method, dataset) cell, the per-seed vector of train, validation, and test accuracies; the hook is a strict no-op when its environment variable is unset. Within DHG-Bench a seed fixes the train/validation/test split for all methods. Consequently, for a fixed dataset, the per-seed test accuracies of any two methods are paired observations on identical splits. This property is what licenses the paired tests used in Section 3.3 and is not available from the reported summary statistics alone.

A wall-clock budget of one hour per cell and the 40 GB memory limit were enforced. Cells exceeding either were recorded as missing and were not imputed. Of the 260 (method $\times$ dataset) cells, 225 completed. Eighteen cells were lost on the trivago dataset, whose failure modes are described below; only HNHN and LEGCN complete on trivago. The remaining 17 missing cells fall on the three largest datasets (coauthor_dblp, walmart-trips-100, and yelp) and reflect resource limits: 13 to CUDA out-of-memory on the GPU, 2 to host-memory (numpy) allocation failures, and 2 to the one-hour timeout. The affected methods are the memory- and compute-heavy HJRL, DPHGNN, SheafHyperGNN, HyperND, PhenomNN, HyperGT, and the tensor-based TMPHN, together with the clique-expansion baselines CEGCN and CEGAT and, on yelp, LEGCN. The reported DHG-Bench results show out-of-memory in essentially the same heavy-method cells on its 24 GB GPU, so the pattern of missingness in our recompute is consistent with the benchmark's own evaluation. Seven further cells were truncated by the one-hour budget rather than lost: the per-seed hook writes accuracies incrementally, so when the budget expired mid-run, the completed seeds were preserved. The truncated cells are TMPHN on ModelNet40, actor, amazon, pokec, pubmed and twitch, and HyperGCN on yelp, each retaining two of the twenty seeds. They are counted among the 225 completed cells and enter the mean-based layers (Layers 1 and 3) through their two-seed means; the per-seed layers (Layer 2 and the pooled count of Section 4.5) use only the cells with all twenty seeds, and a sensitivity check for the across-datasets layer is reported in Section 4.3.

The trivago dataset required separate handling. Training fails on 18 of the 20 methods: fourteen abort with the device-side assertion $t \geq 0 \ \&\& \ t < n_classes$ in the loss; three (EHNN, HJRL, TMPHN) exhaust host memory while materialising a dense $172,738 \times 405,940$ incidence matrix (261–522 GiB); and DPHGNN exceeds the 40 GB GPU budget. Only HNHN and LEGCN complete. The label tensor shipped with the dataset is itself dense (160 classes indexed $0, \dots, 159$; verified directly on the released data), so the assertion cannot be attributed to sparse label identifiers, and an environment-gated dense-label remap, added as a candidate fix, is a no-op on these labels and does not restore training. The assertion at a dense-label tensor points to an inconsistency between the released data pipeline and the training configuration rather than to a repairable data fault;

we did not trace the fault further, since the dataset is excluded either way, and the verified failure taxonomy is recorded in Appendix B. The two methods that complete fall below their reported values, HNHN severely so (10.3% against 53.8%) and LEGCN moderately (28.4% against 33.5%), against the 33–91% reported upstream across the twelve methods with a reported trivago value. Together these findings indicate that the trivago pipeline distributed with the public repository does not reproduce the configuration that produced the reported numbers. We therefore excluded trivago from the node-classification analysis and report the failure among the findings of the reproducibility audit and the limitations of this study.

3.3. Statistical Analysis

We applied four layers of analysis to the recomputed per-seed accuracies. Throughout, for a fixed dataset, n denotes the number of seeds ($n = 20$); for across-datasets comparisons, k denotes the number of methods, and N denotes the number of datasets entering the comparison.

3.3.1. Layer 1: Reproducibility Check

For every cell present both in our recompute and in the test accuracies reported by DHG-Bench [10] (its node-classification results table and supplementary results), we computed the absolute difference $|\Delta| = |\bar{a}_{\text{ours}} - \bar{a}_{\text{paper}}|$ between mean test accuracies, in percentage points (pp). We summarised the distribution of $|\Delta|$ across cells and reported the fractions falling within the conventional reproducibility tolerances of 1 pp and 3 pp, together with the cells exceeding them.

3.3.2. Layer 2: Per-Dataset Paired Significance

For each dataset and each unordered pair of methods (a, b) present on it, we compared the n paired per-seed test accuracies with the two-sided Wilcoxon signed-rank test [18]. The statistic is $W = \min(R^+, R^-)$, where R^+ and R^- are the rank sums of the positive and negative seed-wise differences. Within each dataset we controlled the family-wise error rate over all $\binom{k_d}{2}$ method pairs (k_d being the number of methods present on dataset d) using the Holm step-down procedure [20] at $\alpha = 0.05$. This per-dataset paired layer is the analysis absent from the original benchmark; it is well posed precisely because each seed fixes a common split (Section 3.2). Cells truncated by the compute budget (Section 3.2) are excluded from this layer, so k_d counts only the methods with all twenty seeds on dataset d .

3.3.3. Layer 3: Across-Datasets Comparison

This layer was restricted to the complete block of $N = 9$ datasets on which all $k = 17$ HNNs return a value (ModelNet40, NTU2012, actor, amazon, coauthor_cora, cora, pokec, pubmed, and twitch), using each method's per-dataset mean accuracy. Following the framework of Demšar [7], methods were ranked on each dataset (rank 1 = best) and compared with the Friedman test [33]. With $\bar{R}_j$ the average rank of method j ,

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^k \bar{R}_j^2 - \frac{k(k+1)^2}{4} \right]. \quad (1)$$

Because the Friedman χ_F^2 is conservative, we also report the Iman–Davenport correction [22],

$$F_F = \frac{(N-1) \chi_F^2}{N(k-1) - \chi_F^2}, \quad (2)$$

distributed as F with $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom. Agreement of the rankings across datasets was measured with Kendall's coefficient of concordance,

$$W_K = \frac{\chi_F^2}{N(k - 1)} \in [0, 1]. \quad (3)$$

When the omnibus test rejected the equality of mean ranks, we located the differences with two complementary post hoc analyses. The Nemenyi test [19] declares two methods as different when their average ranks differ by at least the critical difference

$$CD = q_\alpha \sqrt{\frac{k(k + 1)}{6N}}, \quad (4)$$

with q_α the studentised-range quantile; the outcome is visualised as a critical-difference diagram [7]. Because the Nemenyi procedure is unstable and low-powered for the numbers of methods and datasets considered here [9], we additionally performed all $\binom{k}{2}$ pairwise Wilcoxon signed-rank tests on the per-dataset mean vectors (paired by dataset, $N = 9$) and controlled the family-wise error rate with Holm [8,20]. A pair was reported as significant only when it survived Holm correction at $\alpha = 0.05$.

3.3.4. Layer 4: Variance Decomposition

Following the variance-aware evaluation of Bouthillier et al. [6], for each dataset d , we contrasted the spread between methods with the noise within methods. Let $s_{\text{within}}^{(m)}$ be the seed-wise standard deviation of method m on dataset d and let μ_m be its seed-mean accuracy. We defined the within-method noise as the average $\bar{s}_{\text{within}} = k_d^{-1} \sum_m s_{\text{within}}^{(m)}$, the between-method spread as the standard deviation s_{between} of the means $\{\mu_m\}$, and reported the ratio

$$\rho_d = \frac{s_{\text{between}}}{\bar{s}_{\text{within}}}. \quad (5)$$

A value of $\rho_d \gg 1$ indicates that the differences between methods exceed seed-to-seed jitter, so the dataset can in principle order methods; $\rho_d \lesssim 1$ indicates the opposite. All standard deviations use the unbiased $(n - 1)$ estimator.

3.3.5. Significance-Aware Leaderboard (Complementary View)

As a pooled summary, we counted, over all datasets and all method pairs present on each, the fraction of head-to-head comparisons whose per-seed paired Wilcoxon test reached $p < 0.05$ before multiplicity correction. This aggregate is reported as a descriptive complement to Layers 2 and 3, not as the headline result: pooling uncorrected tests inflates the apparent number of distinguishable pairs, and the quantity is given only to express the per-dataset picture as a single number.

3.4. Structural Characterisation of the Datasets

To move beyond treating each dataset as a black box, we related the per-dataset behaviour of the recompute to the structural properties of the underlying hypergraphs. The structural statistics are taken from the benchmark's own dataset documentation (Table A1 of DHG-Bench [10]) and the source-dataset papers, and we re-derived them independently from the released data to confirm that the characterisation rests on the audited release. The node counts, hyperedge counts, feature dimensionalities, mean hyperedge sizes and class counts reproduce that table exactly, and the hyperedge homophily reproduces it under the intra-class node-pair convention on eleven of the twelve datasets (within 0.02); coauthor_cora is the single exception, where the tabulated value (0.78) lies between the intra-class-pair (0.72) and dominant-label (0.88) conventions. The rankability–homophily association reported below is unchanged whether it is computed on the tabulated or the in-

independently re-derived homophily (Spearman $r = +0.58$ against $r = +0.57$), so it does not depend on the homophily convention or on this single value. For each node-classification dataset we used the number of nodes, the number of hyperedges, the feature dimensionality, the mean hyperedge size $|e|$, the number of classes, and the hyperedge homophily ratio $\mathcal{H}_{\text{edge}}$ introduced by Li et al. [32], which measures the tendency of nodes in a hyperedge to share a label and is high for homophilic and low for heterophilic hypergraphs.

We linked these properties to two recompute-derived quantities defined earlier: the rankability ρ_d of Section 4.4, which states whether a dataset can order methods at all, and the leading architecture family on each dataset, read off the per-dataset mean accuracies. Methods were grouped first by the benchmark's own taxonomy, spectral-based, spatial-based, and tensor-based [10], and then by a finer mechanism reading within those categories (multiset or equivariant set-function aggregation, decoupled training-free message passing, sheaf diffusion, spectral convolution, two-stage vertex–hyperedge propagation, line or clique expansion, nonlinear or energy-based diffusion, transformer attention, and tensor message passing). Associations between a structural property and ρ_d were summarised with the Spearman rank correlation across the twelve analysed datasets; with $N = 12$, these are descriptive associations used to frame the discussion, not significance claims, and they are reported as such; the structural features are themselves correlated, in that the homophilic citation and co-authorship datasets also tend to carry more classes, so the associations are not independent and no multiplicity correction is applied to descriptive statistics of this kind. The structural table, the per-dataset leading family, and these associations are regenerated by the released analysis script, and the rankability values are identical to those in Section 4.4.

3.5. Implementation and Reproducibility

The analyses were implemented in Python. Wilcoxon, Friedman, and Iman–Davenport computations used `scipy.stats`; Holm correction used `statsmodels`; the Nemenyi critical-difference diagram used `scikit-posthocs`. Both source patches (the per-seed dump and the dense-label remap) are environment-gated, so the upstream code is unchanged when the corresponding variables are unset. The analysis code, the patches, the 225 per-seed result files, and the reproducibility-check table are released (Data Availability Statement). The per-cell accuracies, paired-test counts, Kendall's W , and the critical-difference diagram are regenerated by re-running the two analysis scripts on the released per-seed results.

4. Results

We report the four analysis layers in turn, computed on the recomputed twenty-seed accuracies. Throughout, “significant” means surviving Holm correction at $\alpha = 0.05$ unless noted otherwise.

4.1. Reproducibility Against the Reported Means

Across the 222 cells comparable with the test accuracies reported by DHG-Bench [10], the absolute difference between our mean test accuracy and the reported value has a median of 1.8 pp, but the mean is 6.7 pp; 32% of cells fall within 1 pp and 64% within 3 pp (Table 1). The gap between the median and mean signals that the discrepancies are concentrated rather than uniform, and they separate cleanly along two axes: one method and one group of datasets.

The method axis is HJRL. It deviates by 88.8 pp on ModelNet40 and by 83.8, 49.2 and 43.8 pp on NTU2012, Cora-CA and cora, with a mean absolute difference of 34.5 pp over its nine comparable cells, and reproduces no cell within 1 pp. The gap is large, one-directional,

and present across unrelated datasets, which points to a method-specific failure rather than a transcription error.

Because our recompute runs on a newer software stack than the benchmark, this gap could, in principle, be a versioning effect rather than a property of the method. We tested that directly. Rebuilding the upstream environment named by the benchmark (Python 3.9.21, PyTorch 2.2.2+cu121, against our Python 3.12.3 and PyTorch 2.4.0+cu121) and re-running HJRL on the four datasets with the largest deviations reproduces our own recompute exactly: all twenty per-seed accuracies are identical on every dataset, and HJRL still reaches only 34.9% on cora, 34.5% on Cora-CA, 4.3% on NTU2012 and 7.5% on ModelNet40, against the 78.7, 83.7, 88.2 and 96.3% reported. HGNN, run as a control in the same environment, likewise reproduces our recompute exactly and stays within 1.2 pp of its reported value on cora and Cora-CA, confirming that the rebuilt environment trains normally. The HJRL shortfall is therefore not a versioning conflict: it is present under the software stack the benchmark itself specifies, and it is a property of the released method implementation or of the default configuration under which the public release runs it. The full control is reported in Appendix C.

The dataset axis remains after HJRL is removed. Excluding it leaves the median at 1.8 pp but the mean at 5.5 pp, because the residual error is not spread evenly over datasets (Table 2, Figure 2). On the citation and social graphs (pokec, twitch, amazon, coauthor_dblp, cora and coauthor_cora) the median absolute difference is at or below 1.8 pp, and the signed gap is small. On four datasets the recompute is systematically and substantially lower than reported: NTU2012 (mean signed gap −14.8 pp), walmart-trips-100 (−11.6 pp), pubmed (−9.5 pp) and ModelNet40 (−7.4 pp), with actor and yelp weaker in the same direction. On these datasets almost every method underperforms its reported value by a similar margin, and the seed-to-seed standard deviation within each affected cell is small (typically 1–3 pp), so the shortfall is shared across methods rather than driven by a few unstable runs.

A shared, one-directional shortfall across nearly all methods on a dataset is not consistent with each method failing independently; it is consistent with a difference in configuration. The public harness exposes a single default hyperparameter set per method together with a dataset-specific section for cora only; the dataset-specific sections that the documentation refers to are absent for the other datasets (Section 3.2). Our recompute therefore evaluates every method in the common default configuration, whereas the reported numbers on the visual-object datasets NTU2012 and ModelNet40 and on walmart-trips-100 evidently rely on settings not contained in the public release. Independent results in the original method literature reinforce this reading: the journal version of HGNN reports about 96.8% on ModelNet40 and 83.1% on NTU2012 under per-dataset tuning [34], far above our default-configuration recompute on the same datasets, which is consistent with the shortfall reflecting absent configuration rather than an error in the reported values. This is itself a reproducibility finding: the published per-dataset accuracies on these datasets cannot be reproduced from the distributed code and configuration alone. We therefore read the absolute accuracies on NTU2012, ModelNet40 and walmart-trips-100 as a lower-configuration baseline, not as a contradiction of the reported numbers, and we return to the consequences for the leaderboard analysis in Section 5. The full per-cell table is provided as Supplementary Material.

Table 1. Reproducibility of the twenty-seed recompute against the test accuracies reported by DHG-Bench [10], summarised over all comparable cells. Differences are absolute, in percentage points (pp). HJRL is reported separately because it does not reproduce under the modern software stack.

Subset	Cells	Mean $ \Delta $	Median $ \Delta $	≤ 1 pp	≤ 3 pp
All cells	222	6.7	1.8	32%	64%
Excluding HJRL	213	5.5	1.8	33%	65%
HJRL only	9	34.5	28.0	0%	22%

Table 2. Per-dataset reproducibility of the twenty-seed recompute against DHG-Bench [10], for the comparable cells (excluding HJRL and trivago), ordered by signed gap. The signed gap is the recompute minus the reported value, so negative values mean the recompute is lower.

Dataset	Cells	Median $ \Delta $ (pp)	Signed Gap (pp)	≤ 3 pp
pokec	19	0.5	−1.0	89%
twitch	19	0.9	−0.8	100%
amazon	19	1.3	−0.6	100%
coauthor_dblp	16	1.3	−2.0	88%
cora	19	1.8	−2.0	74%
coauthor_cora	19	1.8	−2.4	74%
actor	19	3.3	−6.8	47%
yelp	10	5.2	−4.5	30%
ModelNet40	19	3.8	−7.4	47%
pubmed	19	1.8	−9.5	58%
walmart-trips-100	13	5.1	−11.6	38%
NTU2012	19	12.4	−14.8	21%

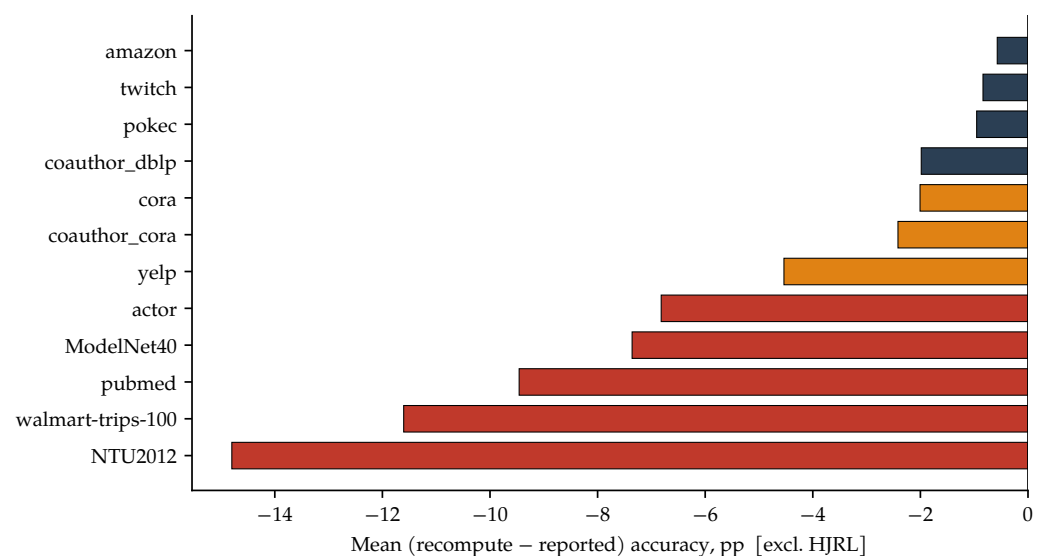


Figure 2. Mean signed reproducibility gap per dataset (recompute minus reported accuracy, in percentage points), excluding HJRL and trivago. Citation and social graphs reproduce closely; the visual-object datasets NTU2012 and ModelNet40 and the co-purchasing dataset walmart-trips-100 show a large shared shortfall, consistent with the absence of dataset-specific configurations in the public release rather than with method-level failure. Colour marks the magnitude band of the plotted gap (largest shortfalls in red, intermediate in orange, smallest in dark blue); it repeats the information already carried by bar length and has no independent meaning.

4.2. Per-Dataset Significance Across Seeds

With twenty seeds, the paired Wilcoxon test distinguishes most pairs of methods on most datasets (Table 3): 98% of pairs on walmart-trips-100 and 96% on coauthor_dblp,

and 74–94% across ModelNet40, actor, pubmed, NTU2012, coauthor_cora and cora. The share falls to 64–70% on pokec, yelp and amazon, and collapses to 21% (36 of 171 pairs) on twitch. The per-seed protocol therefore has adequate power to separate methods on every dataset except twitch (Figure 3), which directly addresses the power concern that a five-seed protocol leaves open. The cause of the twitch exception is identified in Section 4.4.

Table 3. Per-dataset paired Wilcoxon test across seeds ($n = 20$), with Holm correction within each dataset. “Distinguishable” counts the method pairs as significant at $\alpha = 0.05$. Trivago is excluded (Section 3.2), and so are the seven cells truncated by the compute budget, so “methods” counts the methods with all twenty seeds on each dataset.

Dataset	Methods	Distinguishable Pairs	Share
walmart-trips-100	16	118/120	98%
coauthor_dblp	17	130/136	96%
ModelNet40	19	161/171	94%
actor	19	154/171	90%
pubmed	19	150/171	88%
NTU2012	20	165/190	87%
coauthor_cora	20	160/190	84%
cora	20	140/190	74%
pokec	19	119/171	70%
yelp	9	24/36	67%
amazon	19	109/171	64%
twitch	19	36/171	21%

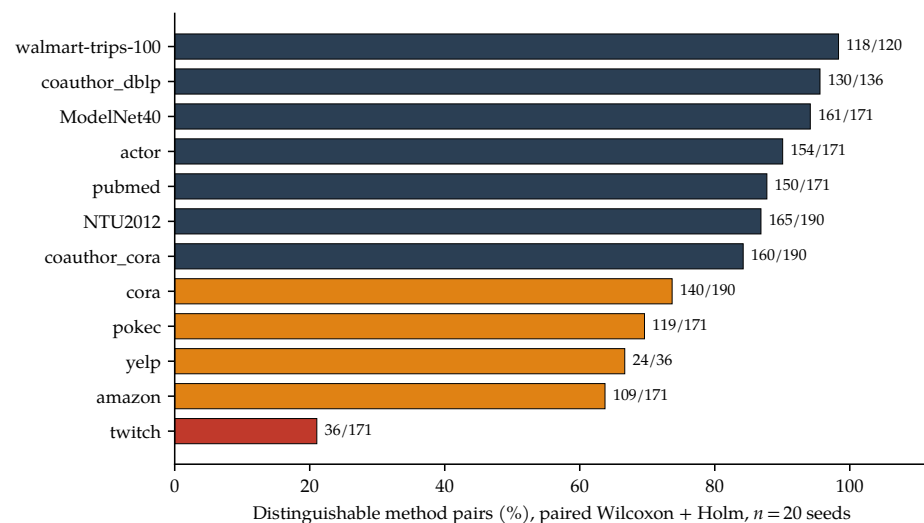


Figure 3. Fraction of method pairs that are statistically distinguishable per dataset (paired Wilcoxon signed-rank test across $n = 20$ seeds, Holm-corrected, $\alpha = 0.05$), over the cells with all twenty seeds. Most datasets separate the large majority of pairs; twitch separates only 21%. Colour marks the magnitude band of the plotted share and repeats the information carried by bar length; it has no independent meaning.

4.3. Across-Datasets Comparison

On the complete block of nine datasets where all 17 HNNs return a value, the Friedman test rejects the equality of mean ranks ($\chi^2_F(16) = 64.4$, $p = 9.4 \times 10^{-8}$; Iman–Davenport $p = 1.9 \times 10^{-10}$). Kendall’s coefficient of concordance is $W_K = 0.447$, indicating moderate agreement of the per-dataset rankings. The average-rank ordering places EDHNN (4.56), AllSetformer (4.67) and EHNN (5.00) at the top, followed by TFHNN (5.56) and SheafHyperGNN (5.67), and HJRL (16.44) last (Figure A1). Six of the 153 cell means entering this block are computed from the two seeds retained by the compute budget (all belonging to

TMPHN; Section 3.2). As a sensitivity check, excluding TMPHN from the block entirely leaves the picture unchanged: Kendall's W moves from 0.447 to 0.445, no pair of the remaining 16 methods survives Holm correction (0 of 120), and the same methods form the statistically inseparable top band.

Despite the significant omnibus result, none of the $\binom{17}{2} = 136$ method pairs survives the Holm-corrected pairwise Wilcoxon test: 0 of 136 pairs are significant (Figure 4). The critical-difference diagram (Figure A1) shows why: the Nemenyi critical difference is larger than the rank gap separating the leading methods, so the entire top group falls inside a single band. The diagram itself is placed in Appendix A rather than in the main text: the Nemenyi procedure is the one component of this analysis that the study itself treats as unstable and low-powered for these numbers of methods and datasets [9], and a critical-difference diagram displayed alongside the primary result invites readers to anchor their takeaways on it. The pairwise conclusion of this section rests on the Holm-corrected Wilcoxon tests, and the diagram is a compact visual summary of the average ranks. The omnibus test therefore establishes that the 17 methods are not collectively equivalent, but no individual method can be declared better than another across these datasets, and, in particular, the top five are statistically indistinguishable; their per-seed accuracies overlap substantially on a representative dataset (Figure 5).

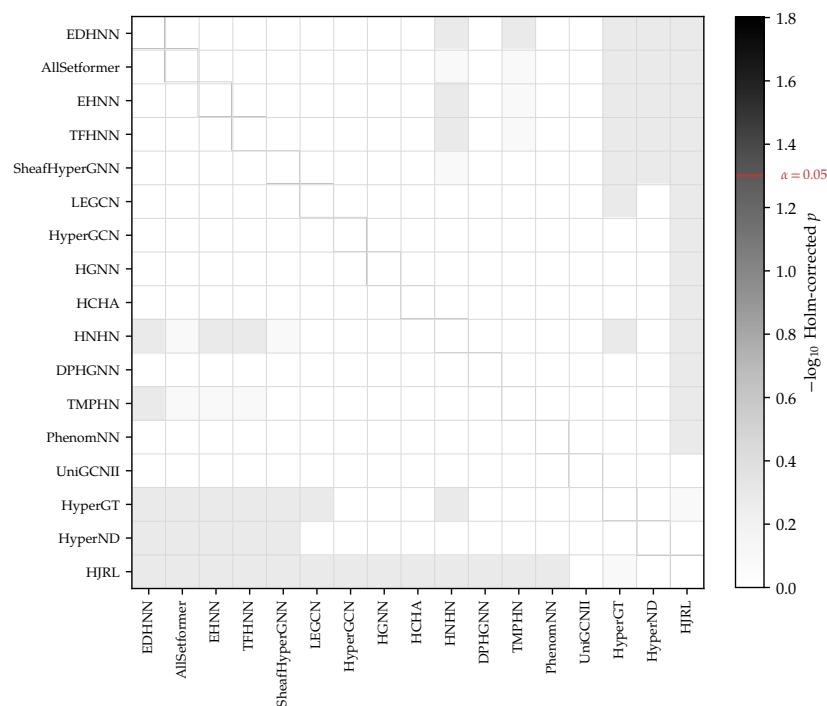


Figure 4. Holm-corrected pairwise Wilcoxon p -values, as $-\log_{10} p$, for all $\binom{17}{2} = 136$ method pairs over the nine complete-block datasets (paired by per-dataset mean, $N = 9$). Methods are ordered by average rank. The line on the colour bar marks $\alpha = 0.05$; no cell reaches it, so no method pair is separable after correction.

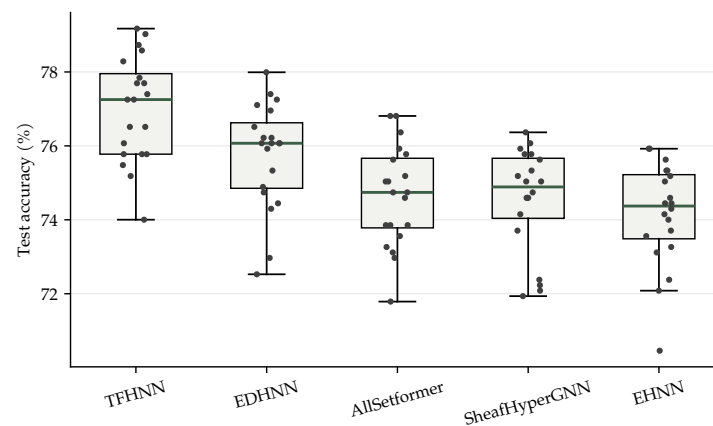


Figure 5. Per-seed test accuracy ($n = 20$) of the five top-ranked methods on cora, ordered by mean accuracy. The distributions overlap substantially with the per-dataset counterpart of the single critical-difference band in Figure A1.

4.4. Variance Decomposition

The ratio ρ_d of between-method spread to within-method seed noise varies widely across datasets (Table 4, Figure 6). On walmart-trips-100 and ModelNet40 the spread between methods is 19 and 12 times the seed noise; on NTU2012, actor, coauthor_cora, coauthor_dblp, cora and pubmed, it lies between 5 and 9. On twitch the ratio is 0.8: the typical seed-to-seed jitter (SD 0.74 pp) exceeds the spread of the method means (SD 0.58 pp). Twitch is therefore structurally unable to order methods, which explains its 21% distinguishable-pair rate in Section 4.2; the limitation is a property of the dataset, not of the test. Amazon (1.8) and yelp (1.5) are weak in the same direction.

Table 4. Variance decomposition per dataset, from the twenty-seed recompute. “Seed SD” is the mean within-method standard deviation across seeds; “Between-Method SD” is the standard deviation of the method means; ρ_d is their ratio. Datasets are ordered by ρ_d ; trivago is excluded.

Dataset	Seed SD (pp)	Between-Method SD (pp)	ρ_d
walmart-trips-100	0.64	12.23	19.2
ModelNet40	1.70	20.74	12.2
actor	1.06	9.43	8.9
NTU2012	2.67	20.64	7.7
coauthor_dblp	0.51	3.47	6.8
coauthor_cora	1.57	10.52	6.7
cora	1.54	9.40	6.1
pubmed	2.43	12.43	5.1
pokec	1.02	2.36	2.3
amazon	0.69	1.24	1.8
yelp	1.32	1.95	1.5
twitch	0.74	0.58	0.8

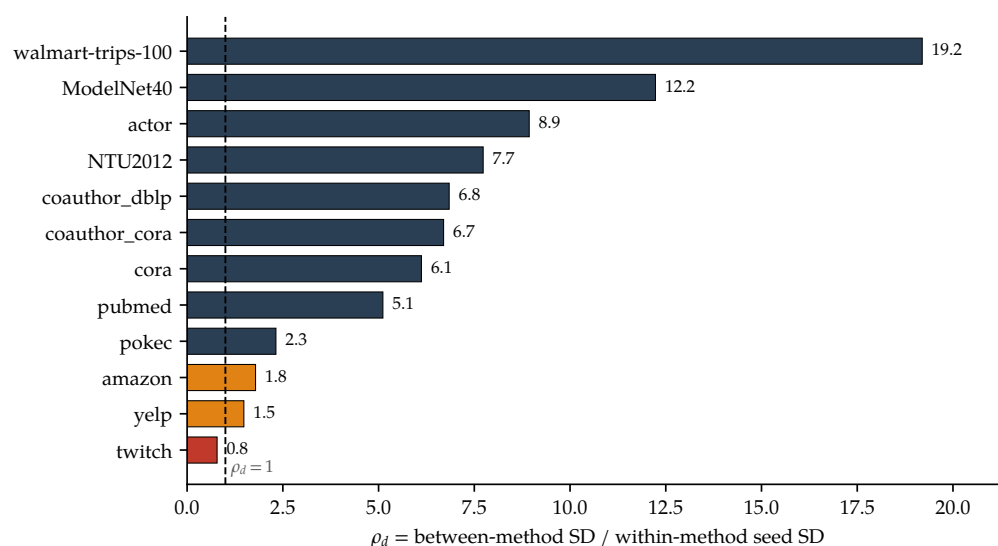


Figure 6. Per-dataset ratio ρ_d of between-method spread to within-method seed noise (twenty-seed recompute). The dashed line marks $\rho_d = 1$, below which seed-to-seed noise exceeds the gap between methods; twitch ($\rho_d = 0.8$) is the only dataset that cannot order methods, with amazon and yelp the next weakest. Colour marks the weakest-ranking datasets and repeats the information carried by bar length; it has no independent meaning.

4.5. Significance-Aware Leaderboard

Pooling all head-to-head comparisons across datasets and pairing by seed, over the cells with all twenty seeds, 1678 of 1889 (89%) reach $p < 0.05$ before multiplicity correction. This pooled figure aggregates the per-dataset picture of Section 4.2: within a dataset, methods are usually separable. It is reported for completeness only. Because it ignores both multiplicity and the across-datasets structure, it must not be read as evidence of a global ranking, which Section 4.3 shows does not survive correction.

4.6. Hypergraph Structure, Rankability, and Leading Family

The variance decomposition shows that datasets differ sharply in whether they can order methods at all, and the across-datasets layer shows that the nominal leader changes from dataset to dataset. Both observations become interpretable once each dataset is described by the structural properties of its hypergraph rather than treated as an opaque benchmark entry. Table 5 pairs the published structural statistics with the rankability ρ_d and the leading hypergraph neural network of the recompute, ordered by ρ_d .

Two structural properties track rankability. Across the twelve datasets, ρ_d rises with the hyperedge homophily ratio (Spearman $r = +0.58$) and with the number of classes ($r = +0.57$), with a weaker association to the clique-expansion density ($r = +0.52$); the feature dimensionality, the mean hyperedge size, and the node count show no monotone association ($|r| \leq 0.18$). The mechanism behind the homophily association is visible in the spread of the methods themselves: the range of mean accuracies across HNNs grows with $\mathcal{H}_{\text{edge}}$ ($r = +0.71$). Figure 7 places the datasets in the homophily–rankability plane. The four datasets that rank methods weakly or not at all sit together at the bottom: twitch ($\rho_d = 0.8$), yelp (1.5), amazon (1.8) and pokec (2.3) are the heterophilic, few-class datasets ($\mathcal{H}_{\text{edge}} \leq 0.49$; twitch and pokec are binary), on which, as the benchmark itself reports, most HNNs fall back towards a feature-only baseline [10]. When every method collapses onto the same near-baseline accuracy, the between-method spread falls below the seed noise, and the dataset can no longer separate methods. Twitch is the extreme case: with a seven-dimensional feature vector, it has the lowest feature dimensionality of any dataset, which limits the signal available to every method, and the methods are indistinguishable

in consequence. The instructive exception is actor, which is heterophilic ($\mathcal{H}_{\text{edge}} = 0.46$) yet ranks methods strongly ($\rho_d = 8.9$): there, the methods do not collapse together but diverge, with the spectral-convolution HGNN and HCHA falling to about 64–67% while the line-expansion and set-based leaders hold near 86%, so the spread is large for the opposite reason.

Table 5. Hypergraph-structural properties of the node-classification datasets (from Table A1 of DHG-Bench [10]; $\overline{|e|}$ is the mean hyperedge size and $\mathcal{H}_{\text{edge}}$ is the hyperedge homophily ratio of [32]), paired with the recompute-derived rankability ρ_d (Section 4.4) and the leading HNN on each dataset. Datasets are ordered by ρ_d ; trivago is excluded. The structural columns were independently re-derived from the released data and reproduce it (exactly for the counts and sizes; within 0.02 for $\mathcal{H}_{\text{edge}}$, taken here under the intra-class node-pair convention), with coauthor_cora the only homophily value to differ.

Dataset	Nodes	Feat.	$\overline{ e }$	$\mathcal{H}_{\text{edge}}$	Classes	ρ_d	Leading HNN (Family)
walmart-trips-100	88,860	100	6.59	0.60	11	19.2	EDHNN (set-based)
ModelNet40	12,311	100	5.00	0.87	40	12.2	EDHNN (set-based)
actor	16,255	50	5.25	0.46	3	8.9	LEGCN (line-expansion)
NTU2012	2,012	100	5.00	0.79	67	7.7	SheafHyperGNN (sheaf)
coauthor_dblp	41,302	1425	4.45	0.87	6	6.8	AllSetformer (set-based)
coauthor_cora	2,708	1433	4.28	0.78	7	6.7	UniGCNII (two-stage)
cora	2,708	1433	3.03	0.75	7	6.1	HCHA (spectral-conv.)
pubmed	19,717	500	4.35	0.78	3	5.1	AllSetformer (set-based)
pokec	14,998	65	2.29	0.45	2	2.3	LEGCN (line-expansion)
amazon	22,299	111	3.10	0.37	5	1.8	EHNN (set-based)
yelp	50,758	1862	6.66	0.29	9	1.5	EDHNN (set-based)
twitch	16,812	7	6.23	0.49	2	0.8	AllSetformer (set-based)

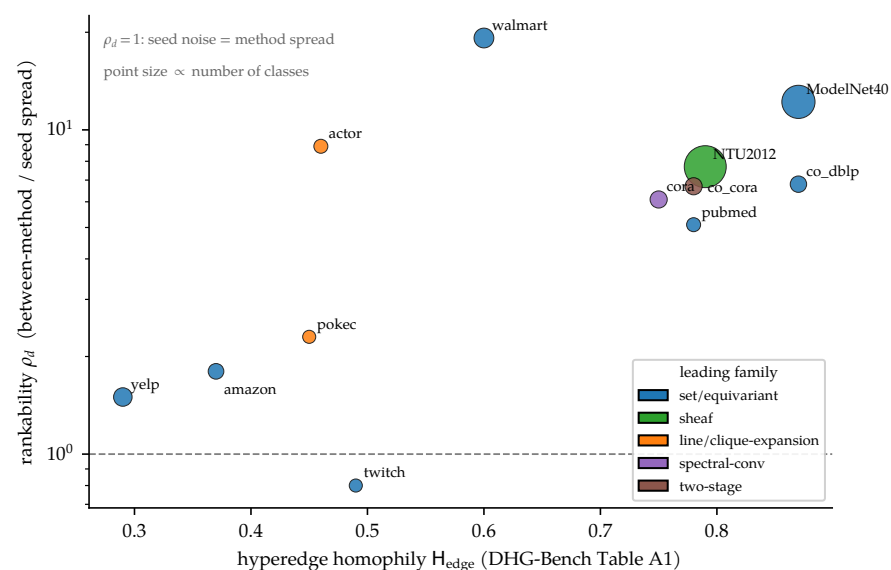


Figure 7. Rankability ρ_d against hyperedge homophily $\mathcal{H}_{\text{edge}}$ for the twelve analysed datasets (point size proportional to the number of classes; colour marks the leading family). The dashed line is $\rho_d = 1$, below which seed noise exceeds the between-method spread. Rankability increases with homophily and class count; the heterophilic, few-class datasets cluster below, with twitch the only dataset that cannot order methods.

Which family leads also tracks structure, but only at a finer resolution than the benchmark's own taxonomy. Averaged over the per-dataset mean ranks, the benchmark's three categories are almost level: tensor-based 7.29, spatial-based 7.79, and spectral-based 8.95, a span of under two rank positions, and the spectral category contains both a top method (TFHNN) and the worst (HJRL). The coarse label is therefore a poor predictor of perfor-

mance. A mechanism-level grouping separates the methods far more clearly: the multiset and equivariant set-function methods (AllSetformer, EDHNN, EHNN) lead at an average mean rank of 4.28, ahead of the decoupled training-free TFHNN (5.75) and the sheaf method (5.91); the line- and clique-expansion methods (7.55) and the spectral-convolution methods (8.11) sit in the middle; and the transformer (11.73), the nonlinear-diffusion methods (11.86) and the non-reproducible cross-expansion HJRL (16.44) trail. The per-dataset leaders in Table 5 follow this picture: the set-based methods lead on seven of the twelve datasets, concentrated on the homophilic and larger-hyperedge cases where higher-order aggregation is rewarded, whereas on the heterophilic actor and pokec, the line-expansion LEGCN leads, consistent with the benchmark's finding that naive expansion baselines and a feature-only MLP rival the dedicated HNNs once homophily is lost [10]. These family-rank differences are descriptive: across the nine complete-block datasets, no method pair survives Holm correction (Section 4.3), and the leading set-based, training-free and sheaf methods fall within the single critical-difference band. The families are assigned from the methods' published mechanisms, detailed in the Supplementary Material, rather than from the observed ranks; several of the finer families (the sheaf and transformer groups) contain a single method, so their family rank is that method's rank rather than a generalisation across an architecture class. The structural reading therefore explains the within-dataset wins without contradicting the across-datasets inseparability, and we develop both points in Section 5.

5. Discussion

5.1. Within-Dataset Separability Versus Across-Datasets Ranking

The two significance layers point in opposite directions, and the contrast is the central result of this study. Within a single dataset, twenty seeds are enough to separate the large majority of method pairs: the Holm-corrected paired Wilcoxon test distinguishes 74–98% of pairs on most datasets (Section 4.2). The per-seed protocol therefore has ample power, and the concern that a five-seed evaluation might simply be underpowered does not survive a larger seed count on these datasets. Across datasets, however, the same machinery returns no significant pair at all: over the complete block of nine datasets, the Friedman test rejects the equality of mean ranks, yet none of the 136 method pairs survives Holm correction, and the leading methods fall inside a single critical-difference band (Section 4.3).

These statements are not in tension once the unit of analysis is made explicit. A within-dataset test asks whether method *a* beats method *b* on that dataset; an across-datasets test asks whether *a* beats *b* consistently over a heterogeneous suite. The first question has a clear answer on almost every dataset; the second does not, because the ordering of methods changes from dataset to dataset. Kendall's $W_K = 0.447$ quantifies this directly: the per-dataset rankings agree only moderately, which is exactly the regime in which an omnibus test can reject global equality while no single pair is separable after correction. A leaderboard that reports one number per method and orders by it suppresses this disagreement and presents a stability that the data do not support.

5.2. Implications for Leaderboard Claims

The practical consequence concerns how benchmark leaders are described. The average-rank ordering in our recompute places EDHNN first, ahead of AllSetformer, EHNN and TFHNN, but the top five methods are statistically indistinguishable from one another (Section 4.3). The benchmark under audit highlights a different method as a consistent top performer [10]; our recompute puts a different method nominally first and, more importantly, shows that neither claim is supported once multiplicity is taken into account. The disagreement about which method is “best” is itself evidence for the main point: small

changes (a different seed count, a different software stack, a recompute) reshuffle the nominal leader within a band of methods that significance testing cannot separate. This echoes findings in the wider graph-learning literature, where rankings have proven sensitive to the evaluation protocol, and well-tuned baselines often match more elaborate models [1,2,5], and it is consistent with the variance-aware view that between-method differences are frequently small relative to the noise of the experimental pipeline [6].

We do not read this as a claim that the methods are equivalent or that the benchmark is uninformative. The omnibus test does reject global equality, so structure exists; the variance decomposition shows that several datasets discriminate methods strongly (Section 4.4). The narrower and better-supported statement is that the *ordering at the top* is not robust, and that reporting a single leader without a significance analysis overstates what the experiment establishes.

5.3. Datasets That Cannot Rank Methods

The variance decomposition adds a complementary diagnostic that does not depend on any particular test. On twitch the between-method spread is smaller than the within-method seed noise ($\rho_d = 0.8$), so the ranking of methods on that dataset is dominated by the choice of seed rather than by the methods themselves. This explains, without reference to statistical power, why twitch separates only 21% of pairs while every other dataset separates the majority: the dataset simply does not contain enough signal to order the methods. Amazon and yelp are weak in the same direction. A dataset of this kind contributes noise to an aggregate leaderboard while appearing, in a means-only table, to contribute an equally weighted vote. Screening datasets by ρ_d before aggregating is a cheap precaution, and reporting it alongside accuracy would let readers discount datasets that cannot support a ranking.

5.4. Why Performance Varies, and When Families Succeed or Fail

The across-datasets instability is not arbitrary once datasets are read through their structure (Section 4.6). Two regimes account for most of the behaviour. On homophilic hypergraphs with moderate-to-large hyperedges, the visual-object datasets, the co-citation and co-authorship networks, and walmart-trips-100, the higher-order signal is informative, the methods spread out, and the multiset and equivariant set-function methods lead. This is the regime these architectures were designed for: AllSetformer, EDHNN and EHNN aggregate each hyperedge as a permutation-invariant multiset, which preserves the group structure that homophilic hyperedges carry, and they head the average ranking accordingly. The decoupled training-free TFHNN and the sheaf method stay with them inside the same band, so the top of the leaderboard is a small cluster of set-aware and diffusion-aware designs rather than a single architecture.

On heterophilic hypergraphs the picture inverts, and it does so in a way that explains the apparent failures. When nodes joined by a hyperedge tend not to share a label ($\mathcal{H}_{\text{edge}} \lesssim 0.5$), aggregating over hyperedge members mixes classes and erodes the feature signal, so dedicated higher-order message passing offers little over a feature-only model. The benchmark observes directly that most HNNs fall to or below a MLP on its heterophilic datasets [10], and our recompute shows the structural consequence: on actor and pokec, the line-expansion LEGCN, which reduces the hypergraph to a graph before learning, leads the dedicated HNNs, and the spectral-convolution methods, which rely on a global hypergraph eigenstructure that heterophily makes uninformative, lose the most ground. The transformer HyperGT and the heavier diffusion and tensor methods trail in both regimes and additionally exhaust the 40 GB budget on the largest datasets, so their poor average rank reflects a combination of fit and scalability rather than fit alone. The non-

reproducible HJRL is a separate case, treated in Section 4.1. None of these family-level statements is a significance claim: across datasets, the leading methods are statistically inseparable (Section 4.3), and the structural reading explains which method wins a given dataset, not that any family is reliably the best.

5.5. Partial Heuristics for Model Selection

The structural reading supports a few practical heuristics, offered with the explicit caveat that they select a band of near-equivalent methods rather than a guaranteed winner (Table 6). First, inspect rankability before reading a leaderboard: on a dataset with $\rho_d \lesssim 2$, recognisable from low hyperedge homophily, few classes and low feature dimensionality, the order of methods is dominated by the seed rather than by the methods, and the appropriate choice is the cheapest scalable model rather than the nominal leader. Second, on homophilic hypergraphs with moderate hyperedges, a set-function method (AllSetformer, EDHNN) or the training-free TFHNN is a safe top-band choice. Third, on heterophilic hypergraphs ($\mathcal{H}_{\text{edge}} \lesssim 0.5$), expect the dedicated HNNs to sit near a feature-only MLP, treat a strong MLP or a line- or clique-expansion model as a competitive and inexpensive baseline, and be wary of spectral-convolution methods, which are the most prone to collapse. Fourth, on large or memory-bound hypergraphs, prefer the decoupled training-free design, which remained scalable where the heavier diffusion, transformer and tensor methods ran out of memory. These heuristics are empirical and partial; they follow from a single benchmark under its default configuration, and the across-datasets analysis shows that the methods they recommend cannot be statistically separated within the top band. They are best read as hypotheses to test on a new dataset rather than as prescriptions.

Table 6. Partial, empirical model-selection heuristics tied to the structural regime of a dataset. Each recommendation selects a statistically inseparable top band (Section 4.3), not a guaranteed single winner; the absolute accuracies on NTU2012, ModelNet40 and walmart-trips-100 reflect the default configuration (Section 5.7).

Structural Regime	Recommended Family	Caveat/Why
Homophilic, moderate $ \overline{e} $ (citation, co-authorship, visual; $\mathcal{H}_{\text{edge}} \gtrsim 0.6$)	set-function (AllSetformer, EDHNN) or training-free TFHNN	top band; ρ_d high, so the dataset does rank methods
Heterophilic ($\mathcal{H}_{\text{edge}} \lesssim 0.5$)	strong MLP or line/clique-expansion (LEGCN); avoid spectral-convolution	HNNs sit near a feature-only MLP; higher-order aggregation can hurt
Low rankability ($\rho_d \lesssim 2$; few classes, low feature dimensionality, e.g., twitch)	cheapest scalable model (TFHNN); do not read the order	ranking is seed-dominated; methods statistically tied
Large-scale/memory-bound (walmart, yelp, trivago)	decoupled/training-free TFHNN	out-of-memory prone: PhenomNN, DPHGNN, HyperGT, tensor methods

5.6. Reproducibility Findings

The recompute surfaced three reproducibility issues of different character. First, HJRL does not reproduce from the public release: its accuracies fall tens of percentage points below the reported values across several unrelated datasets, in a consistent direction, and a controlled re-run under the upstream software stack named by the benchmark reproduces our own numbers exactly (Section 4.1, Appendix C), so the gap is a property of the released method implementation or its default configuration rather than a versioning effect or a reporting error. Second, and more consequential for users of the benchmark, the public release does not ship the dataset-specific configurations that the reported numbers rely on.

The configuration files expose a global default and a single dataset-specific section for cora; for the visual-object datasets NTU2012 and ModelNet40 and for walmart-trips-100, every method run in the default configuration falls short of its reported accuracy by a similar, dataset-wide margin (Section 4.1). Because the shortfall is shared across nearly all methods and the within-cell seed variance is small, the most plausible explanation is a difference in configuration rather than independent method failures. The published per-dataset accuracies on these datasets therefore cannot be reproduced from the distributed code and configuration alone, a gap that matters precisely because the benchmark's value to the community is as a reusable, runnable resource. Third, training on trivago fails for 18 of the 20 methods in the public repository, through a label-range assertion for fourteen and memory exhaustion for four, even though the released label tensor is dense; a dense-label remap is consequently a no-op and does not restore training, and the two methods that complete fall below their reported accuracies, indicating that the distributed data pipeline does not reproduce the configuration behind the reported numbers.

None of these issues invalidates the across-datasets conclusion. The complete-block analysis is restricted to the nine datasets on which all methods return a value; the result is that *fewer* differences are significant, and a recompute that systematically lowered some accuracies would, if anything, compress the methods further rather than manufacture separation. The reproducibility findings are reported as findings in their own right: they document where the public benchmark can and cannot be reproduced, which is information a benchmark user needs and which a means-only leaderboard does not convey.

5.7. Interpreting Results Under the Default Configuration

Because the recompute evaluates every method in the single configuration the public harness exposes, the consequences of that choice should be stated in one place. The distributed configuration files contain a global default section and a dataset-specific section for cora alone; the per-dataset sections that the documentation refers to are absent for the other twelve datasets (Section 3.2). The default set is therefore the only configuration reproducible from the public release, and it is the configuration under which every method is compared here. Its effect is confined and is now well localised: on three datasets, the visual-object pair NTU2012 and ModelNet40 and the co-purchasing walmart-trips-100, every method falls short of its reported accuracy by a similar, dataset-wide margin, which the external anchor of HGNN+ under per-dataset tuning (about 96.8% on ModelNet40 and 83.1% on NTU2012 [34]) confirms is a configuration gap rather than a reporting error. On the remaining datasets the recompute reproduces the reported means closely. The absolute accuracies on these three datasets are accordingly a lower-configuration baseline, not a contradiction of the published numbers. The per-dataset configurations behind the reported numbers are not part of the public release as accessed; we audit that public, runnable release as the object of study, and recovering tuned or privately held configurations would change what is being audited rather than settle whether the release reproduces its own table. We have nevertheless contacted the DHG-Bench authors through a public issue in the benchmark repository, requesting the per-dataset configuration sections behind the reported numbers; any configurations released in response will be incorporated in the follow-up work named in the conclusions.

This default-configuration ceiling does not reach the conclusions of this study, and the reason is structural rather than incidental. The significance and variance analyses operate within a dataset and across seeds: the per-dataset paired tests (Section 4.2), the rankability ρ_d (Section 4.4) and the structural reading (Section 4.6) compare methods that all run under the same configuration on that dataset, so a shared, dataset-wide shift in absolute accuracy leaves the comparisons, and the rankings they produce, unchanged. The across-

datasets analysis is by construction conservative: a recompute that lowers some accuracies compresses the methods rather than separating them, so it can only reduce the number of significant pairs, which is already zero. Tuned per-dataset configurations would raise the absolute accuracies on the three affected datasets and might sharpen within-dataset separability, but the across-datasets instability arises from the disagreement of rankings between datasets and from within-method seed noise, neither of which a uniform per-dataset retuning is positioned to remove. The default configuration therefore qualifies the absolute numbers on three datasets and leaves the rankability, the variance decomposition and the central no-single-leader conclusion intact.

A further question is whether the default configuration could bias the variance decomposition itself: if the configuration shortfall meant that every method failed to converge on the affected datasets, the methods would be compressed towards a common floor, and ρ_d would understate how strongly those datasets discriminate between methods. The recompute allows a direct check, and the observed pattern is the opposite. The three datasets with the large configuration shortfall are the three most rankable in this study (walmart-trips-100 $\rho_d = 19.2$, ModelNet40 12.2, NTU2012 7.7), and they separate 87–98% of method pairs in the per-seed tests, whereas the compressed datasets (twitch $\rho_d = 0.8$, yelp 1.5, amazon 1.8, pokec 2.3) are among those that reproduce the reported means most closely (signed gaps between -0.6 and -4.5 pp). The reason is that the shortfall is shared in direction but not in magnitude: methods lose unequal amounts of accuracy under the default configuration, which widens the between-method spread rather than narrowing it, and the small within-cell seed variation on the affected datasets (typically 1–3 pp) shows each method converging stably to its default-configuration level rather than failing to converge. The compression observed on the heterophilic, few-class datasets is structural, with most methods falling back towards a feature-only baseline, and not configurational. The default configuration therefore does not artificially compress the variance decomposition on these data; if anything, it inflates ρ_d on the three affected datasets, and the caveat recorded in the limitations, that tuned configurations might change within-dataset separability, remains the boundary of the claim.

5.8. A Reusable Protocol

The four layers used here (a reproducibility check, per-dataset paired testing across seeds, an across-datasets omnibus with post hoc analysis, and a variance decomposition) are not specific to hypergraph learning. They require only per-seed scores under a common split, which any benchmark can emit, and they are implemented with standard libraries. We release the analysis code and the per-seed results so that the same significance-aware view can be attached to other benchmarks. The recommendation that follows from this study is modest and concrete: report per-seed scores, test the ranking rather than only displaying it, screen datasets that cannot discriminate methods, and treat the top of a leaderboard as a band rather than an order unless significance testing shows otherwise.

5.9. Limitations

Several limitations qualify these conclusions. This study covers the node-classification track only; the hyperedge-prediction and hypergraph-classification tracks of the benchmark may behave differently and are left to future work. The across-datasets analysis is restricted to the nine datasets on which all 17 methods complete, so methods that are absent on the largest datasets due to out-of-memory or timeout are compared on a smaller suite; the missing cells are concentrated on the memory- and compute-heavy methods, and a larger memory budget than the 40 GB used here would close some of them. The absolute accuracies on NTU2012, ModelNet40 and walmart-trips-100 reflect the default configuration

rather than tuned per-dataset settings and should not be read as the best achievable performance of each method on those datasets; as set out in Section 5.7, this default-configuration ceiling is confined to the absolute numbers on these three datasets and leaves the rankability, the variance decomposition and the no-single-leader conclusion intact, because the significance and structural analyses compare methods within a dataset and across seeds under a shared configuration. The Nemenyi critical-difference diagram is known to be unstable and low-powered for the numbers of methods and datasets involved [9], which is why the headline pairwise conclusion rests on the Holm-corrected Wilcoxon tests rather than on the diagram; the diagram is accordingly provided in Appendix A, as a visualisation rather than the primary test. Finally, the analysis inherits whatever dataset-construction choices are built into the public harness, including the hypergraph construction for the visual-object datasets; we audit reproducibility against the reported numbers and the released code, not the upstream design decisions themselves.

6. Conclusions

We recomputed the node-classification track of a recent hypergraph-learning benchmark independently, with twenty seeds and a modern software stack, and applied a four-layer statistical audit to the per-seed results. The picture that emerges is consistent across layers. Given enough seeds, methods are usually distinguishable within a single dataset, so the per-seed protocol is not underpowered. Across datasets, the leaderboard does not survive significance testing: an omnibus test rejects the global equality of the 17 methods, but no pair is separable after Holm correction, and the leading methods lie within a single critical-difference band. The nominal top method changes between the original report and our recompute, which is itself a symptom of an unstable top rather than a correction to it.

The variance decomposition shows why aggregation is fragile: at least one benchmark dataset carries more seed-to-seed noise than between-method signal and therefore cannot order methods at all, while contributing an equally weighted vote to a means-only leaderboard. The recompute additionally documents a method that does not reproduce under a modern stack, a public release that does not ship the per-dataset configurations its reported numbers depend on, and a dataset whose distributed pipeline does not reproduce its reported accuracies. These are reproducibility findings in their own right, and none of them manufactures the across-datasets conclusion, which is, by construction, conservative.

The contribution is not a new model but a reusable way of reading a benchmark. Reporting per-seed scores, testing a ranking rather than displaying it, screening datasets that cannot discriminate methods, and treating the top of a leaderboard as a band unless significance shows otherwise are inexpensive practices that any benchmark can adopt. Relating the per-dataset behaviour to hypergraph structure adds an actionable layer on top of the audit: rankability and the leading architecture family track the hyperedge homophily and the class granularity of a dataset, set-function methods leading where higher-order structure is informative and expansion methods or a feature-only baseline becoming competitive once homophily is lost, which yields partial model-selection heuristics even though the methods within the top band remain statistically inseparable. We release the analysis code and the per-seed results so that the same significance-aware view can be attached elsewhere. Future work includes extending the audit to the hyperedge-prediction and hypergraph-classification tracks, recovering the missing heavy-method cells under a larger memory budget, and assessing whether tuned per-dataset configurations change the within-dataset separability while leaving the across-datasets instability in place.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/make8080224/s1>, Table S1: Per-cell reproducibility of the twenty-seed recompute against the accuracies reported by DHG-Bench, for every method–dataset cell; Table S2: Per-dataset structural features, rankability and leading method, in machine-readable form; Table S3: Assignment of each method to its published mechanism family; Table S4a: Leading method on each dataset with its family, mean accuracy and rankability; Table S4b: Mean rank of each mechanism family across datasets; Table S5: Spearman correlations between each structural feature and rankability; Table S6: Independent re-derivation of the structural features from the released data; Table S7: Inventory of the seven cells truncated by the compute budget.

Author Contributions: This work was conducted within the framework of the Bauman Scientific Collaborators (BSC)—Computer Science Research Group; all authors are members or collaborators of the group, and the study design and the evaluation-rigour questions were coordinated at the group level. Individual contributions in the Contributor Roles Taxonomy (CRediT) are as follows. Conceptualisation, V.V.T., S.O.K. and V.S.T.; methodology, S.O.K. and V.S.T.; software, S.O.K.; validation, V.V.T., V.S.T., A.S.B., A.H. and T.C.; formal analysis, S.O.K. and A.S.B.; investigation, S.O.K. and A.S.B.; resources, V.V.T., V.S.T. and A.H.; data curation, S.O.K. and A.S.B.; writing—original draft preparation, S.O.K.; writing—review and editing, V.V.T., S.O.K., V.S.T., A.S.B., A.H. and T.C.; visualisation, S.O.K.; supervision, V.V.T. and V.S.T.; project administration, V.S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The analysis code, the two source patches, the 225 per-seed result files, and the reproducibility-check table that support the findings of this study are openly available in the project repository at <https://github.com/digger32/significance-hypergraph> (accessed on 5 June 2026). The benchmark under audit, DHG-Bench, and the datasets it uses are third-party resources available from their original authors; this study redistributes neither. No new experimental data were generated beyond the recomputed per-seed accuracies released in the repository.

Acknowledgments: The authors acknowledge the organisational and scientific support of the Bauman Scientific Collaborators (BSC)—Computer Science Research Group. During the preparation of this manuscript, the authors used a generative AI tool (Anthropic Claude, versions Opus 4.8 and Fable 5) in two supporting roles: (1) language editing of this manuscript for grammar, style, and consistency; and (2) assistance in drafting the LaTeX code for the tables, figures, and document structure and the Python code for the released analysis scripts. All research questions, the experimental design, the recomputation and statistical analysis, the interpretations, and the conclusions are the work of the authors. Every AI-assisted suggestion was reviewed and verified by the authors, who take full responsibility for the entire content of this manuscript. No AI tool was used to fabricate data or references; all numerical results were produced by the released analysis code, and no AI-generated text appears verbatim in the scientific claims or conclusions.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Critical-Difference Diagram

Figure A1 shows the Nemenyi critical-difference diagram for the 17 hypergraph neural networks over the nine complete-block datasets, computed on the twenty-seed recompute. It is provided in this appendix rather than in the main text because the Nemenyi test is unstable and low-powered for the numbers of methods and datasets considered here [9]; the primary pairwise analysis of Section 4.3 is the Holm-corrected Wilcoxon testing. The diagram is retained as a visual summary of the average ranks and of the single band that joins the leading methods.

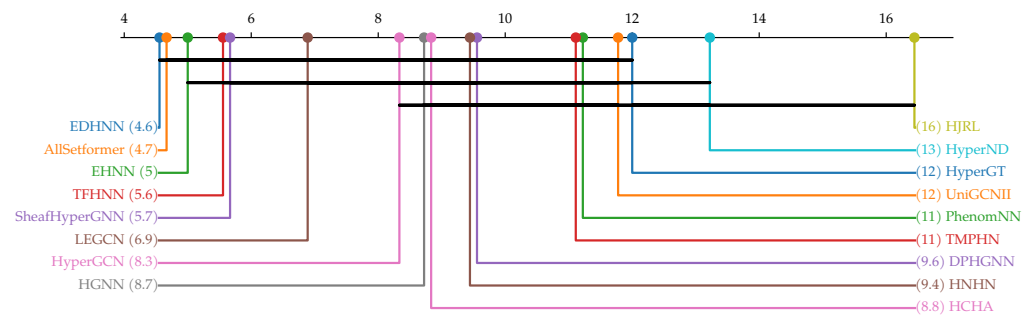


Figure A1. Critical-difference diagram (Nemenyi, $\alpha = 0.05$) for the 17 HNNs over the nine complete-block datasets, from the twenty-seed recompute. Methods are placed by average rank (lower is better); horizontal bars join groups whose average ranks differ by less than the critical difference. The top cluster lies within a single band.

Appendix B. The Trivago Failure Mode

This appendix records the verified failure taxonomy of the trivago dataset under the public release, as a reference for users attempting to run or repair the pipeline. Table A1 classifies all twenty methods. Fourteen abort at the first training step with the device-side assertion $t \geq 0 \ \&\& \ t < n_classes$ raised inside the cross-entropy loss; three exhaust host memory while materialising the incidence structure as a dense matrix; one exceeds the GPU budget; two complete.

The label tensor shipped with the dataset was probed directly on the released data: it contains 160 classes indexed $0, \dots, 159$, with the number of distinct labels equal to the maximum index plus one, so the labels are already dense and the environment-gated dense-label remap of Section 3.2 is a no-op (verified with the remap enabled and disabled; the probe output is identical). An assertion of this form at a dense, in-range label tensor implies that some component downstream of the loader operates with a class count smaller than 160, for example, a model head built with fewer outputs, which is an inconsistency between the released data pipeline and the training configuration rather than a repairable fault in the label encoding. The memory failures are of independent origin: EHNN, HJRL and TMPHN attempt to materialise the $172,738 \times 405,940$ incidence matrix densely, and DPHGNN forms an attention product beyond any single-GPU budget.

Table A1. Failure taxonomy of the twenty methods on trivago under the public release (one-hour budget, 40 GB GPU). Reported values are those of DHG-Bench [10].

Outcome	Methods	Count	Detail
Completes	HNHN, LEGCN	2	test accuracy $10.29 \pm 0.13\%$ and $28.38 \pm 0.71\%$ against reported 53.8% and 33.5%
Label-range assertion	MLP, HGNN, HyperGCN, HCHA, HyperND, PhenomNN, SheafHyperGNN, TFHNN, UniGCNII, AllSetformer, EDHNN, HyperGT, CEGCN, CEGAT	14	$t \geq 0 \ \&\& \ t < n_classes$ at the first training step
Host-memory exhaustion	EHNN, HJRL, TMPHN	3	dense $172,738 \times 405,940$ incidence; 522, 261 and 261 GiB requested
GPU out-of-memory	DPHGNN	1	111 GiB attention product against the 40 GB budget

The two methods that complete fall below their reported accuracies over the twenty seeds, HNNH severely so. For HNNH the training accuracy ($10.28 \pm 0.09\%$) is indistinguishable from its test accuracy ($10.29 \pm 0.13\%$), so the model does not fit even its own training split, which is consistent with the same pipeline–configuration inconsistency rather than with overfitting or seed noise. Since trivago is excluded from every analysis layer, the fault was not traced further; the run logs and the label probe accompany the released code.

Appendix C. Control Experiment Under the Upstream Software Stack

The recompute of Section 4.1 runs on Python 3.12.3 with PyTorch 2.4.0+cu121, whereas DHG-Bench specifies Python 3.9 and PyTorch 2.2. To establish whether the HJRL shortfall is a consequence of that difference, the upstream environment was rebuilt (Python 3.9.21, PyTorch 2.2.2+cu121, CUDA 12.1, same GPU, same released code, same default configuration, same twenty seeds) and two methods were re-run on the four datasets with the largest HJRL deviations: HJRL itself, and HGNN as a control that reproduces well on citation graphs and therefore indicates whether the rebuilt environment trains normally.

Table A2. Control experiment under the upstream software stack. Accuracies are means over twenty seeds; the two rightmost columns give the difference against the value reported by DHG-Bench [10] and against our recompute on the modern stack. Every per-seed accuracy is identical between the two stacks, so the last column is zero throughout.

Method	Dataset	Reported	PyTorch 2.4	PyTorch 2.2	2.2–Reported	2.2–2.4
HJRL	Cora	78.67	34.90	34.90 ± 3.19	−43.77	0.00
HJRL	Cora-CA	83.72	34.48	34.48 ± 4.25	−49.24	0.00
HJRL	NTU2012	88.15	4.31	4.31 ± 0.75	−83.84	0.00
HJRL	ModelNet40	96.33	7.55	7.55 ± 0.97	−88.78	0.00
HGNN	Cora	77.90	79.08	79.08 ± 1.32	+1.18	0.00
HGNN	Cora-CA	82.84	82.39	82.39 ± 1.44	−0.45	0.00
HGNN	NTU2012	90.13	66.21	66.21 ± 3.14	−23.92	0.00
HGNN	ModelNet40	97.43	88.80	88.80 ± 0.79	−8.63	0.00

Two conclusions follow. First, HJRL fails identically under the stack the benchmark specifies, so its shortfall is not a versioning conflict; the control shows the rebuilt environment is functional, since HGNN reproduces its reported accuracy to within 1.2 pp on both citation graphs. Second, the shortfall of every method on NTU2012 and ModelNet40 is likewise unchanged across stacks (HGNN loses 23.9 and 8.6 pp under both), which is consistent with the missing per-dataset configurations discussed in Section 5.7 rather than with the software environment. The exact agreement between the two stacks is expected here rather than surprising: both builds link the same CUDA 12.1 libraries, the seeds fix the initialisation and the splits, and the operations involved are deterministic on this hardware, so the per-seed trajectories coincide. The control was run twice, in separate sessions, with the interpreter and library versions recorded in the run log.

References

1. Shchur, O.; Mumme, M.; Bojchevski, A.; Günnemann, S. Pitfalls of Graph Neural Network Evaluation. *arXiv* **2018**, arXiv:1811.05868.
2. Errica, F.; Podda, M.; Bacciu, D.; Micheli, A. A Fair Comparison of Graph Neural Networks for Graph Classification. *arXiv* **2020**, arXiv:1912.09893.
3. Dwivedi, V.P.; Joshi, C.K.; Luu, A.T.; Laurent, T.; Bengio, Y.; Bresson, X. Benchmarking Graph Neural Networks. *J. Mach. Learn. Res.* **2023**, *24*, 1–48.
4. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020; pp. 22118–22133.

5. Dehghani, M.; Tay, Y.; Gritsenko, A.A.; Zhao, Z.; Houlsby, N.; Diaz, F.; Metzler, D.; Vinyals, O. The Benchmark Lottery. *arXiv* **2021**, arXiv:2107.07002.
6. Bouthillier, X.; Delaunay, P.; Bronzi, M.; Trofimov, A.; Nichyporuk, B.; Szeto, J.; Mohammadi Sepahvand, N.; Raff, E.; Madan, K.; Voleti, V.; et al. Accounting for Variance in Machine Learning Benchmarks. In Proceedings of the Machine Learning and Systems (MLSys), Virtual, 5–9 April 2021; Volume 3.
7. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30. [[CrossRef](#)] [[PubMed](#)]
8. García, S.; Herrera, F. An Extension on “Statistical Comparisons of Classifiers over Multiple Data Sets” for All Pairwise Comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
9. Benavoli, A.; Corani, G.; Mangili, F. Should We Really Use Post-Hoc Tests Based on Mean-Ranks? *J. Mach. Learn. Res.* **2016**, *17*, 1–10. [[CrossRef](#)] [[PubMed](#)]
10. Li, F.; Wang, X.; Zhang, W.; Zhang, Y.; Lin, X. DHG-Bench: A Comprehensive Benchmark for Deep Hypergraph Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Rio de Janeiro, Brazil, 23–27 April 2026.
11. Chien, E.; Pan, C.; Peng, J.; Milenkovic, O. You Are AllSet: A Multiset Function Framework for Hypergraph Neural Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
12. Luo, Y.; Shi, L.; Wu, X.M. Classic GNNs Are Strong Baselines: Reassessing GNNs for Node Classification. In Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track, Vancouver, BC, Canada, 10–15 December 2024.
13. Platonov, O.; Kuznedelev, D.; Diskin, M.; Babenko, A.; Prokhorenkova, L. A Critical Look at the Evaluation of GNNs under Heterophily: Are We Really Making Progress? In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
14. Lim, D.; Hohne, F.; Li, X.; Huang, S.L.; Gupta, V.; Bhalerao, O.; Lim, S.N. Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–14 December 2021.
15. Bechler-Speicher, M.; Amos, I.; Gilad-Bachrach, R.; Globerson, A. Graph Neural Networks Use Graphs When They Shouldn’t. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*; PMLR: New York, NY, USA, 2024; Volume 235, pp. 3284–3304.
16. Bechler-Speicher, M.; Finkelshtein, B.; Frasca, F.; Müller, L.; Tönshoff, J.; Siraudin, A.; Zaverkin, V.; Bronstein, M.M.; Niepert, M.; Perozzi, B.; et al. Position: Graph Learning Will Lose Relevance Due to Poor Benchmarks. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*; PMLR: New York, NY, USA, 2025; Volume 267, pp. 81067–81089.
17. Kurashkin, S.O.; Tynchenko, V.S.; Borodulin, A.S.; Hammoud, A.; Tee, C. Evaluation Rigor from Graph Neural Networks to Graph Foundation Models: A Systematic Review and a Four-Axis Reporting Standard. *Mach. Learn. Knowl. Extr.* **2026**, *8*, 194. [[CrossRef](#)]
18. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]
19. Nemenyi, P.B. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
20. Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scand. J. Stat.* **1979**, *6*, 65–70.
21. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced Nonparametric Tests for Multiple Comparisons in the Design of Experiments in Computational Intelligence and Data Mining: Experimental Analysis of Power. *Inf. Sci.* **2010**, *180*, 2044–2064. [[CrossRef](#)]
22. Iman, R.L.; Davenport, J.M. Approximations of the Critical Region of the Friedman Statistic. *Commun. Stat.-Theory Methods* **1980**, *9*, 571–595. [[CrossRef](#)]
23. Li, R.; Liu, C. Quantifying Statistical Heterogeneity and Reproducibility in Cooperative Multi-Agent Reinforcement Learning: A Meta-Analysis of the SMAC Benchmark. *Algorithms* **2025**, *18*, 653. [[CrossRef](#)]
24. Antelmi, A.; Cordasco, G.; Polato, M.; Scarano, V.; Spagnuolo, C.; Yang, D. A Survey on Hypergraph Representation Learning. *ACM Comput. Surv.* **2023**, *56*, 24. [[CrossRef](#)]
25. Gao, Y.; Zhang, Z.; Lin, H.; Zhao, X.; Du, S.; Zou, C. Hypergraph Learning: Methods and Practices. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2548–2566. [[CrossRef](#)] [[PubMed](#)]
26. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2019; Volume 33, pp. 3558–3565. [[CrossRef](#)]
27. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P. HyperGCN: A New Method for Training Graph Convolutional Networks on Hypergraphs. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
28. Dong, Y.; Sawin, W.; Bengio, Y. HNHN: Hypergraph Networks with Hyperedge Neurons. In Proceedings of the ICML Graph Representation Learning and Beyond (GRL+) Workshop, Virtual, 13–18 July 2020.
29. Huang, J.; Yang, J. UniGNN: A Unified Framework for Graph and Hypergraph Neural Networks. In Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), Virtual, 19–26 August 2021; pp. 2563–2569. [[CrossRef](#)] [[PubMed](#)]

30. Wang, P.; Yang, S.; Liu, Y.; Wang, Z.; Li, P. Equivariant Hypergraph Diffusion Neural Operators. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
31. Wang, F.; Pena-Pena, K.; Qian, W.; Arce, G.R. T-HyperGNNs: Hypergraph Neural Networks via Tensor Representations. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 5044–5058. [[CrossRef](#)] [[PubMed](#)]
32. Li, M.; Gu, Y.; Wang, Y.; Fang, Y.; Bai, L.; Zhuang, X.; Liò, P. When Hypergraph Meets Heterophily: New Benchmark Datasets and Baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2025; Volume 39, pp. 18377–18384. [[CrossRef](#)]
33. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]
34. Gao, Y.; Feng, Y.; Ji, S.; Ji, R. HGNN+: General Hypergraph Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3181–3199. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.