GRAPH STRUCTURAL ENCODINGS FROM k-FWL ALGEBRA: OPERATIONS, BUDGETS, AND READOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structural encodings supply graph statistics to learning models, yet the field lacks a common language for comparing them: a new channel family is proposed, a gain is reported, and it is rarely clear whether the gain reflects new structural information or only better access to information already present. This paper supplies that language for k-dimensional folklore Weisfeiler–Leman refinement (k-FWL). On a fixed finite graph we show that each refinement round is generated exactly by pointwise operations and contractions whose replacement branches share a single pivot, and we build three dictionary families on this operation: polynomial, endpoint, and joint channels. Each family carries an explicit budget (degree Q, tuple order k, retained objects m, moment P) with sufficient refinement-round membership bounds, so constructions are compared in a common currency rather than by channel count. We then separate what an encoding contains from what a readout can see: channelwise powers are injective on exact vectors and multisets, adding no raw distinction, yet the same expansion changes what mean pooling observes, and every joint triple first moment collapses to n^{-2} on equal-size regular graphs of minimum degree at least one. Graph-pair experiments on BREC and within-graph experiments on ZINC, QM9, OGB, LRGB, and TU graphs evaluate nodes, edges, and vertex pairs against automorphism orbits at matched width; endpoints combined with powers improve pair discrimination on four TU datasets with 95% intervals above zero, while node and edge changes are smaller and can reverse. We report unresolved comparisons alongside averages so that high capacity is not mistaken for full recovery. Operations, budgets, readouts, and object domains jointly determine the distinctions an encoding exposes, and greater discrimination alone does not establish a downstream benefit.

1 INTRODUCTION

Graph structural encodings are now standard in graph learning, yet the field lacks a common language for comparing them: a new channel family is proposed, a gain is reported, and it is rarely clear whether the gain reflects new structural information or only better access to information already present. Random-walk, subgraph, homomorphism, and simple-path statistics enrich graph representations (Ma et al., 2023; Bouritsas et al., 2023; Bao et al., 2025; Airale et al., 2025), and algebraic approaches connect expressive power to polynomial features and tensor operations (Maron et al., 2019; Geerts & Reutter, 2022; Puny et al., 2023); what these constructions lack is a shared currency in which they can be placed side by side.

In practice, an encoding must select finitely many statistics for the objects a model processes, and the same degree budget yields widely different channel counts across constructions. An expressivity characterization of a complete family leaves the choice of finite channels and of readout open. We ask: *under a finite budget, which computations add distinctions, and which of those distinctions remain available after readout?* Increasing matrix powers, exposing endpoint summaries, and counting through a common pivot are different choices, and their outputs behave differently under a multiset readout and a mean. A larger vector can expose statistics to pooling even when every added channel is determined by channels already present, so the final predictor alone does not identify these effects.

We answer this question for a specified family of deterministic encodings, which we call *shared-pivot dictionaries*, and we take k-dimensional folklore Weisfeiler–Leman refinement (k-FWL) as

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

our reference. We prove that k-FWL refinement is generated exactly by pointwise algebra and contractions whose replacement branches share one pivot (Maron et al., 2019; Feng et al., 2023): sharing the pivot records how several vertices relate to the same intermediate vertex, whereas independent branch summaries retain only their separate statistics. We use this operation to *derive* three dictionary families (polynomial, endpoint, and joint channels) whose budgets and relation to the reference are explicit; that relation is a sufficient membership bound rather than an exact implementation of iterative k-FWL.

Two effects organize the analysis: separating previously identical feature vectors, and changing what a restricted readout can observe. An improvement after pooling can thus arise from better access to existing information, without separating additional objects.

The empirical study follows the same questions. BREC (Wang & Zhang, 2024) compares encoding operations and readouts on graph pairs; Section 5 reports the archived counts. Object-discrimination experiments then evaluate nodes, edges, and vertex pairs against automorphism orbits. The object domain matters: at matched width, combining powers and endpoints improves pair capacity over powers alone on four TU datasets, whereas node and edge changes are smaller and can reverse. This locates where added channels resolve structural ambiguities; predictive value still depends on the task and model.

Contributions.

1. **A generative algebra for structural encodings.** We prove that k-FWL refinement is generated exactly by pointwise algebra and shared-pivot contractions (Theorem 1), and we *derive* rather than posit the three dictionary families of Section 3.
2. **Budgets as a common currency.** Every dictionary carries an explicit budget (Q, k, m, P) with sufficient refinement-round membership bounds (Proposition 1, Appendix C). A single degree budget can imply channel counts differing by orders of magnitude: at $Q = 8$ the pair seed has 11 channels, while the triple joint block alone has $9^3 = 729$, so channel count is an unreliable proxy for capacity.
3. **Information is not readout.** We show that channelwise powers are injective on feature vectors and on exact multisets, adding no raw distinction, while the same expansion changes what a mean readout can observe; we also derive a concrete collapse, that every joint triple first moment equals n^{-2} on all equal-size regular graphs with minimum degree at least one.
4. **Matched-width, object-domain-resolved measurement.** On ZINC, QM9, OGB, LRGB, and TU graphs we measure node, edge, and pair discrimination against automorphism orbits at matched width. Mixed exceeds Wider Powers on the pair domain of ENZYMES, MUTAG, NCI1, and NCI109 with 95% intervals above zero, while node and edge changes are smaller and can reverse. High average capacity is not full recovery: on ZINC subset pairs at $d = 14$, $10{,}735$ of $12{,}000$ graphs retain unresolved comparisons. At unrestricted depth and width, precomputed explicit channels do not enlarge the function class (Proposition 2, Appendix D).

Scope. The algebraic statements concern fixed finite graphs under exact arithmetic; the measurements use the protocols of Sections 5 and 6 and concern structural distinction rather than predictive accuracy. The BREC and PPGN records are archived outputs of the implementation.[1]

2 Related Work

Structural encodings. Distance Encoding represents node sets through graph distances (Li et al., 2020) and Graphormer through shortest-path and centrality encodings (Ying et al., 2021); GSN augments message passing with subgraph isomorphism counts (Bouritsas et al., 2023); MoSE uses homomorphism counts and compares their expressivity with random-walk encodings (Bao et al., 2025); SPSE uses simple-path counts to expose local cyclic patterns (Airale et al., 2025), and 2-FWL architectures combine local connectivity, spectral, and distance encodings (Chen et al., 2025). SAN learns spectral positions (Kreuzer et al., 2021), SignNet and BasisNet handle eigenvector sign and basis ambiguities (Lim et al., 2023), and LSPE learns structural and positional representations separately (Dwivedi et al., 2022a). These families supply constructions and guarantees and are usually

[1] Code available at `https://anonymous.4open.science/r/reproducibility-135D/`.

evaluated end to end; read through the shared-pivot operation of Section 3.1, they become different selections within one algebra. We instead fix that algebra and measure its operations and readouts separately.

Algebraic descriptions of expressivity. WL refinement characterizes important limits of message passing and higher-order networks (Xu et al., 2019; Morris et al., 2019), and comparative analyses cover structural augmentation (Papp & Wattenhofer, 2022) and node-based subgraph hierarchies (Zhang et al., 2023), and redundancy-free message passing propagates each path along a single flow rather than the WL tree (Chen et al., 2022). PPGN uses polynomial maps and matrix products (Maron et al., 2019); tensor languages relate indices and summation depth to WL expressivity (Geerts & Reutter, 2022); G2N2 derives graph networks from an algebraic grammar (Piquenot et al., 2024). Puny et al. (2023) characterize equivariant polynomial bases, analyze their tensor contractions, and add polynomial features and operations to existing networks; this is a direct antecedent of our construction. Generalized FWL neighborhoods provide a broader space of iterative architectures (Feng et al., 2023). We use a round-wise FWL reference to analyze selected finite channels, including redundancies and readout losses, rather than to claim a new general polynomial hierarchy.

Discrimination and prediction. The information an encoding carries and the discrimination of a model that consumes it are distinct objects of analysis (Black et al., 2024), and BREC evaluates realized graph discrimination (Wang & Zhang, 2024). In a multi-model positional-encoding benchmark, Grötschla et al. (2026) find that WL distinguishability does not reliably predict downstream performance. We use direct graph-pair readouts and within-graph automorphism orbits to examine specific encoding changes; the latter describes unresolved object roles and measures neither target relevance nor generalization, and downstream discussion is restricted to the GNNPlus setting (Luo et al., 2025).

Position of this work. Prior work establishes which structures a complete encoding family can represent and reports end-to-end gains. We isolate a narrower question: given a fixed finite budget, which selected channels add distinctions, and which survive a specified readout on a specified object domain. We make the shared-pivot operation explicit, bound each dictionary family by membership rather than completeness, separate raw feature information from readout information with exact examples in both directions, and measure discrimination under matched channel widths across node, edge, and pair domains.

3 Constructing Finite Structural Encodings

We first isolate the operation that drives joint refinement, then derive finite channels from it. All constructions below use simple, undirected, unlabeled graphs. Let A be the adjacency matrix, $n = |V|$, and I, J the identity and all-ones matrices. Throughout, a *budget* is the tuple (Q, k, m, P): a degree range Q, a tuple order k, a retained object count $m = |\Omega(G)|$ induced by the object domain, and a moment budget P. Figure 1 summarizes the construction.

3.1 A Shared Pivot and the FWL Reference

Why a common pivot matters. Compare

$$c_{\mathrm{joint}}(u,v) = \sum_t A_{ut}A_{tv}, \qquad c_{\mathrm{sep}}(u,v) = \left(\sum_t A_{ut}\right)\left(\sum_t A_{tv}\right).$$

The first counts common neighbors; the second equals $\deg(u)\deg(v)$ and discards the overlap between the two neighborhoods. The difference is whether both factors share the same pivot: on a six-cycle, pairs at distances two and three have $c_{\mathrm{sep}} = 4$ but common-neighbor counts one and zero.

For $k \geq 2$, let $\mathbf{v} \in V^k$ and write $\mathbf{v}_{i\leftarrow t}$ for replacement of coordinate i by t. Tuples may contain repeated vertices. The initial color C_0 records tuple equalities and adjacencies. Our k-FWL convention uses the update

$$C_{\ell+1}(\mathbf{v}) = \mathrm{HASH}(C_\ell(\mathbf{v}), \{\{(C_\ell(\mathbf{v}_{1\leftarrow t}), \ldots, C_\ell(\mathbf{v}_{k\leftarrow t})) : t \in V\}\}). \tag{1}$$

Each pivot contributes one joint replacement profile. Because the joint multiset determines the k coordinatewise multisets, this convention refines coordinatewise WL (Feng et al., 2023). Graph comparison uses tuple-color multisets with color identifiers shared across graphs.

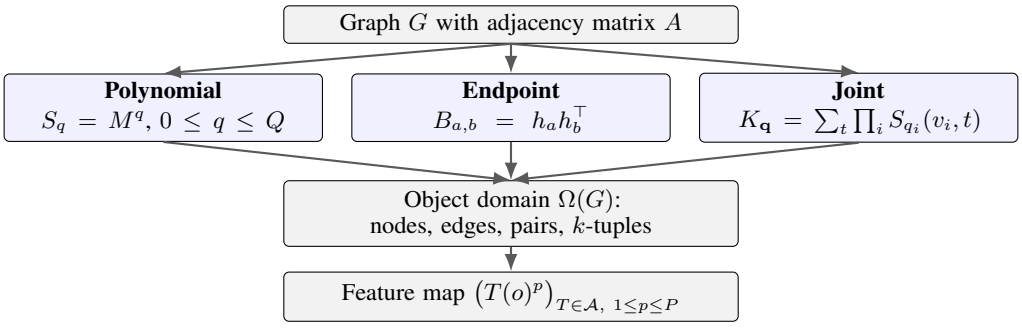


Figure 1: The shared-pivot construction. The three dictionary families share the same operations and differ only in the statistic they retain: weighted walks, endpoint environments, and shared-pivot patterns. All are evaluated on the object domain $\Omega(G)$ under the budget (Q, k, m, P) and expanded channelwise up to the moment budget P.

A channel is a function $T : V^k \to \mathbb{R}$. Define

$$\mathrm{Contract}(T_1, \dots, T_k)(\mathbf{v}) = \sum_t \prod_{i=1}^{k} T_i(\mathbf{v}_{i\leftarrow t}). \tag{2}$$

Applied to color indicators, this contraction counts pivots with a specified joint profile. These counts and the old color determine the next refinement class. For triples, each input T_i is a third-order array, and $\mathrm{Contract}(T_1, T_2, T_3)(u, v, w) = \sum_t T_1(t, v, w)T_2(u, t, w)T_3(u, v, t)$.

Let W_ℓ be the span of the round-ℓ color indicators; each W_ℓ is closed under pointwise products. Let Alg denote closure under finite linear combinations and pointwise products, including the constant function. Set $H_0 = W_0$ and

$$H_{\ell+1} = \mathrm{Alg}(H_\ell \cup \{\mathrm{Contract}(T_1, \dots, T_k) : T_i \in H_\ell\}).$$

Theorem 1 (partition equivalence)**.** *For every fixed finite graph, $k \ge 2$, and $\ell \ge 0$, $H_\ell = W_\ell$. In other words, these operations generate exactly the functions that are constant within each round-ℓ color class.*

Proof sketch. Expanding contraction inputs in color indicators expresses every output as a linear combination of joint-profile counts. These counts and the old indicators are constant on each next-round class. Conversely, polynomial interpolation over the finitely many attained profiles recovers every next-round indicator. Induction gives the equality; Appendix A gives the proof. □

Membership in H_L is therefore an *upper* bound on discriminative power: a channel constant on round-L classes cannot separate objects that round-L k-FWL leaves in one class. The characterization is partitional on a fixed graph, so its interpolation degrees may depend on that graph and it supplies neither a uniformly small dictionary nor canonical labels across graphs (Appendix A).

3.2 Polynomial, Endpoint, and Joint Channels

The object domain $\Omega(G)$ specifies where values are retained: nodes, existing edges, ordered pairs, or tuples. Node features use matrix diagonals; edge features use existing-edge positions. A dictionary is an ordered list of channels evaluated on that domain.

Polynomial channels. Let D be the degree matrix and set

$$M = D^{-1/2}AD^{-1/2}, \qquad S_q = M^q, \qquad 0 \le q \le Q,$$

with zero inverse degree at isolated vertices and $S_0 = I$. The pair seed dictionary is $(I, J, A, S_1, \dots, S_Q)$. Each $S_q(u, v)$ sums the symmetric-normalization weights of all length-q walks between u and v. For higher-order tuples, we lift each matrix to coordinate pairs, $X_{ij}(\mathbf{v}) = X(v_i, v_j)$, retaining a single constant channel. Polynomial tuple features contain these lifts; lifting alone does not perform a joint contraction.

Endpoint channels. Define $h_a = S_a\mathbf{1}$ and append

$$B_{a,b}(u,v) = h_a(u)h_b(v), \qquad B_{a,b} = S_aJS_b, \qquad a,b \geq 0, \quad 0 < a+b < Q. \tag{3}$$

The matrix identity uses the symmetry of S_b. Here $h_a(u) = \sum_x S_a(u,x)$ is the total weight of length-a walks from u, pooled over their destinations, and $B_{a,b}$ aggregates walks of total length $a+b$; the bound $0 < a+b < Q$ is therefore the walk-length budget matching the polynomial dictionary. These channels expose the endpoint environments, including $B_{a,0}(u,v) = h_a(u)$ and $B_{0,b}(u,v) = h_b(v)$, and evaluation uses only the vectors h_a and their outer products. Higher-order dictionaries use the same coordinate lifts.

Joint channels. For $k \geq 3$, append

$$K_{\mathbf{q}}(v_1,\ldots,v_k) = \sum_t \prod_{i=1}^{k} S_{q_i}(v_i,t), \qquad \mathbf{q} \in \{0,\ldots,Q\}^k. \tag{4}$$

All coordinates refer to the same pivot. For triples, $K_{(1,1,1)}$ counts the common neighbors of three vertices, equivalently the homomorphisms of the three-star $K_{1,3}$ into G with ordered leaves. For monomial seeds, the channel counts weighted walks of the specified lengths meeting at a common endpoint, relating the construction to rooted polynomial and homomorphism statistics (Puny et al., 2023; Bao et al., 2025).

To express equation 4 through the FWL contraction, choose the cyclic permutation $j(i) = i+1$ for $i < k$ and $j(k) = 1$, and let $T_i(\mathbf{x}) = S_{q_{j(i)}}(x_{j(i)}, x_i)$. Then

$$\mathrm{Contract}(T_1,\ldots,T_k)(\mathbf{v}) = \sum_t \prod_i S_{q_{j(i)}}(v_{j(i)},t) = K_{\mathbf{q}}(\mathbf{v}). \tag{5}$$

Each joint channel is thus one contraction of seed lifts, applied to fixed seeds, whereas full FWL contracts newly refined states at every round. For $k = 2$ the same statistic is $S_aS_b^\top = M^{a+b}$ by symmetry of M, so it reduces to an extended polynomial budget and is not a separate pair family.

Relation to refinement rounds. For pairs, equation 2 gives $\mathrm{Contract}(Y,X) = XY$. On a fixed finite graph, degree interpolation places M and its lifts in H_1, and balanced matrix multiplication builds powers through Q in another $\lceil \log_2 Q \rceil$ contraction rounds. Thus the seeds lie in H_L and the endpoint and joint dictionaries in H_{L+1}, where $L = 1 + \lceil \log_2 Q \rceil$. By the upper-bound reading above, no channel in these dictionaries separates tuples that the corresponding k-FWL round leaves together. The bounds are sufficient rather than tight: they neither prove strict separation between the families nor make a finite dictionary complete for FWL. Appendices B and C give the examples, definitions, and proofs. **Budgets and cost.** Q controls the degree range, k the tuple order, and $m = |\Omega(G)|$ the retained object count. Table 1 gives direct dense evaluation bounds for fixed k; restricting the object domain reduces stored features but not the construction of the seed dictionary. Connectivity-guided sparsification instead reduces the cost of the 2-FWL computation itself (Chen et al., 2026).

Table 1: Direct evaluation bounds, additional to the polynomial dictionary for the endpoint and joint families; $e_Q = Q(Q+1)/2 - 1$. These are evaluation bounds, not measured efficiency comparisons.

Family	Statistic	Storage	Time
Polynomial	weighted walks	$O(Qn^2)$	$O(Qn^3)$
Endpoint	endpoint summaries	$O(Qn + me_Q)$	$O(Qn^2 + me_Q)$
Joint	shared-pivot patterns	$O(m(Q+1)^k)$	$O(mn(Q+1)^k)$

4 What the Readout Preserves

Fix an object domain and dictionary $\mathcal{A}$. Raw discrimination compares the feature vectors $E_{\mathcal{A}}(o) = (T(o))_{T\in\mathcal{A}}$. A graph multiset retains the vectors and their multiplicities but drops the shared vertex indexing of their positions. A mean retains selected aggregate statistics, a restricted form of invariant set aggregation (Zaheer et al., 2017). The following statements concern exact arithmetic. We use *readout* for any function of the feature map that is compared across objects; the two readouts studied here are the exact feature multiset and a channelwise mean. Figure 2 separates the two effects analyzed below.

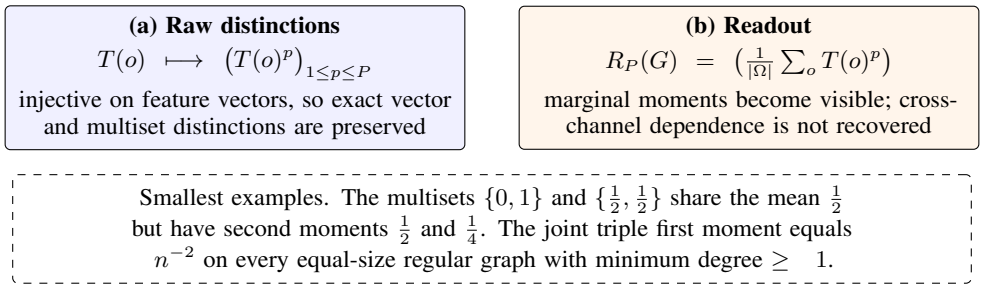


Figure 2: Appending channels has two separate effects: expanding every channel into powers adds no raw distinction (left), yet changes what a restricted mean readout observes (right). The dashed row gives the smallest examples of Section 4.

4.1 Additional Channels and Additional Information

Appending channels while retaining the original vector cannot remove raw distinctions. A strict gain requires at least one new channel to vary on objects whose original vectors agree. In particular, appending a deterministic function of the existing vector preserves exactly its vector and multiset distinctions.

This observation applies within our dictionaries. Since $B_{a,b} = B_{a,0} \odot B_{0,b}$ under the Hadamard product, endpoint products add no raw distinctions once the corresponding single-endpoint channels are retained, and $K_{(a,b,0)}(u,v,w) = S_a(u,w)S_b(v,w)$ is determined by the seed lifts. Not every channel in a joint dictionary represents an independent increase in information. The complete dictionaries retain these channels to specify a common construction and readout; channel count therefore does not equal the number of independent structural statistics.

A complete change of polynomial basis at the same degree budget also preserves raw information, since the associated linear maps are invertible on the complete endpoint and joint blocks; this does not extend to selected channel prefixes or rounded values (Appendix C.4).

4.2 Mean Moments and a Sharp Collapse

For a moment budget P, append the channelwise powers through P and read out

$$R_P(G) = \left(\frac{1}{|\Omega(G)|} \sum_{o\in\Omega(G)} T(o)^p \right)_{T\in\mathcal{A},\, 1\le p\le P}.$$

At each object the expanded feature vector retains its original channels at $p = 1$. Increasing P thus adds no raw vector or multiset distinctions, but exposes higher moments to mean pooling. These channelwise moments omit mixed-channel moments and need not determine the full multiset.

For example, scalar multisets $\{0,1\}$ and $\{1/2,1/2\}$ share the mean $1/2$ but have second moments $1/2$ and $1/4$, so appending squares lets a mean distinguish them without changing the information already present. There is also a limit: $\{(0,0),(1,1)\}$ and $\{(0,1),(1,0)\}$ have identical moments of every order in each individual channel, yet their mean cross-channel products are $1/2$ and zero.

For symmetric seeds, a joint triple channel satisfies

$$\sum_{u,v,w} K_{(a,b,c)}(u,v,w) = \sum_t h_a(t)h_b(t)h_c(t), \qquad h_q = S_q\mathbf{1}. \tag{6}$$

On a regular graph with minimum degree at least one, $M\mathbf{1} = \mathbf{1}$, hence $h_q = \mathbf{1}$. Every joint triple sums to n, so the mean first moment is n^{-2} (n^{-3} after dividing the channel by n). Endpoint channels likewise reduce to J on this class. These joint first moments therefore cannot distinguish equal-size regular graphs. The identity concerns the stated first moments only; the complete joint multiset and the higher moments can still differ.

A linear change of complete polynomial basis commutes with first-moment pooling, so raw information equivalence need not imply equivalence under a restricted nonlinear readout.

4.3 Further Computation by a Backbone

A backbone that retains vertex indexing can access relationships that a direct input multiset has discarded: a six-cycle and two disjoint triangles share the same adjacency-entry multiset, yet $\mathrm{tr}(A^3)$ is 0 and 12 respectively. Conversely, if a backbone can build the explicit channels from the seeds, retain the needed intermediate states, and select the seeds from explicit inputs, then its function class at unrestricted finite depth and width is unchanged, $\mathrm{Rep}(B, E_{\mathrm{seed}}) = \mathrm{Rep}(B, E_{\mathrm{explicit}})$ (Proposition 2, Appendix D). Neither observation establishes that a trained model realizes the available capacity, and fixed-width behavior may differ.

5 Graph Discrimination on BREC

BREC contains 400 non-isomorphic graph pairs (Wang & Zhang, 2024). We use the supplied reference partition, which is computed under the deterministic joint-replacement convention of equation 1: 270 pairs separated by 2-FWL, 110 separated only by 3-FWL, and 20 separated by neither. Pair encodings evaluate the first group; triple encodings evaluate the first two. The 20 neither-group pairs are excluded because every channel of the matching order lies in the algebra H_{L+1} of Section 3, so none can separate them in exact arithmetic; they remain in the benchmark denominator but are not measured failures. The counts reported below are the outputs of the implementation described in Appendix E.1.

5.1 Which Encoding Operations Add Distinctions?

The archived experiment evaluates polynomial, polynomial-plus-endpoint, and endpoint-plus-joint dictionaries at $Q = 2, 4, 8, 16$. It computes the maximum budget once and selects channels by identity for smaller budgets. The archived monomial multiset counts are reproduced in Appendix F. These comparisons fix Q; augmentation increases channel count and computation, so they are not equal-cost comparisons.

Following BREC's convention for deterministic non-GNN methods, direct evaluation compares one stored instance of each graph without the reliable pairwise comparison (RPC) procedure (Wang & Zhang, 2024, Appendix O); features use float32, symmetric normalization, eight-decimal multiset rounding, and a mean tolerance of 10^{-8}, with no relabeling checks (Appendix E.1). On the 270 2-FWL pairs the endpoint dictionary records 262 multiset detections at $Q = 4$ against 211 for the polynomial dictionary, and 270 against 269 at $Q = 16$; at $Q = 8$ the order-3 polynomial dictionary, which has no joint block, records 268 of 270 and 109 of 110. The endpoint increment therefore shrinks as the polynomial dictionary saturates.

5.2 Which Distinctions Survive the Readout?

We compare the feature multiset with mean moments through $P = 1, 2, 4$. The dictionary is fixed within each row of Table 2. Lost and added detections are joined by graph-pair identity, rather than inferred by subtracting totals.

Table 2: Monomial joint-triple readouts, one panel per reference group. Each mean cell is $s\,[\ell, a]$: s detections, ℓ multiset-only, and a mean-only. Rounding before and after aggregation are different comparisons.

2-FWL group (270 pairs)

		Mean		
Q	Multiset	P=1	P=2	P=4
2	210	134 [76, 0]	209 [3, 2]	209 [3, 2]
4	270	191 [79, 0]	224 [46, 0]	238 [32, 0]
8	270	233 [37, 0]	261 [9, 0]	269 [1, 0]
16	270	270 [0, 0]	270 [0, 0]	270 [0, 0]

Only-3-FWL group (110 pairs)

		Mean		
Q	Multiset	P=1	P=2	P=4
2	88	15 [73, 0]	15 [73, 0]	17 [72, 1]
4	108	27 [81, 0]	28 [80, 0]	80 [28, 0]
8	110	42 [68, 0]	76 [34, 0]	105 [5, 0]
16	110	97 [13, 0]	108 [2, 0]	109 [1, 0]

For example, the archive records 108 multiset detections and 80 fourth-order mean detections at $Q = 4$ on the only-3-FWL group. The exact-information analysis predicts no new raw distinctions

from increasing P; the experiment asks what the restricted mean can recover. Mean-only detections cannot establish greater information than the exact multiset. Appendix F.1 retains the integrated scale comparison.

5.3 Recorded Neural Comparison

A separate campaign fits a pair PPGN (Maron et al., 2019) and a third-order shared-pivot extension to each eligible graph pair, asking whether the trained output separates pairs that the recorded input multiset does not. Both models consume the same float32 powers $[I, S, \ldots, S^8]$, where $S = D^{-1/2}AD^{-1/2}$: 9 channels per pair, or 27 channels from the three pair-coordinate lifts per triple. The input baseline, which compares unrounded row multisets, separates 254 of 270 pairs and 379 of 380 triples; source hashes match the archived implementation. At two blocks and width 16, the trained pair model adds 11 to 13 detections per seed, whereas the trained third-order model gains none and loses 12 to 24 on the only-3-FWL group; the two models are not parameter-matched, so this comparison does not separate architecture from capacity. Because the decisions compare float32 features under tolerances, a recorded separation need not reflect a structural distinction; Appendix G gives the architecture, readout, per-seed results, and decision protocol.

6 Object Discrimination on Real Graphs

We measure object discrimination within ZINC (Dwivedi et al., 2023), QM9 (Ramakrishnan et al., 2014), TU (Morris et al., 2020), OGB (Hu et al., 2020), and LRGB (Dwivedi et al., 2022b) graphs. This locates structural ambiguity, without measuring predictive accuracy. Appendix E.2 gives the full dataset sources and versions.

6.1 Metrics and Comparison Scope

Automorphism orbits group the objects a graph symmetry maps to one another, and a topology-only equivariant encoding is constant within each orbit in exact arithmetic. Let T_{orb} contain object pairs from different orbits, S_E the comparisons separated by encoding E, r the orbit count, and α the largest mutually separated object set. We report

$$\mathrm{PR}(E;G) = \frac{|S_E \cap T_{\mathrm{orb}}|}{|T_{\mathrm{orb}}|}, \qquad \mathrm{RC}(E;G) = \frac{\alpha}{r}. \tag{7}$$

Pairwise Resolution (PR) weights cross-orbit object comparisons; Representative Capacity (RC) counts mutually distinguishable representatives relative to the number of orbits. To compute α, we form a similarity graph with one vertex per object and an edge between matching feature vectors; its maximum independent set then selects a largest mutually distinguishable set. PR can be close to one while RC still registers missing representatives. We average both metrics over eligible graphs with equal graph weights; RC is not the fraction of graphs fully recovered.

We evaluate nodes, unordered edges, and unordered pairs of distinct vertices. The width scan selects $d = 6, 8, 10, 12, 14$ channels per orientation from Powers or Endpoints; Mixed concatenates both prefixes, whereas Wider Powers and Wider Endpoints each use $2d$ channels, so Mixed versus Wider Powers is an equal-width comparison. These prefixes differ from the complete Q-bounded dictionaries of Section 3, and Appendix E.3 specifies the channel ordering, normalization, tolerances, orbits, and maximum-independent-set computation. Paired comparisons use common eligible graphs with reliable features and exact capacity solutions, with 95% graph-bootstrap confidence intervals (CI). Distance-gated RC values above one use a different metric (Appendix K).

6.2 Additional Distinctions Depend on the Object Domain

Representative capacity is high on most node and edge domains and lower on several pair domains (Appendix K); a six-cycle has one node orbit but three pair orbits, so node resolution need not imply pair resolution.

Pair-domain gains survive width matching. At $d = 14$, Mixed exceeds Wider Powers on pairs in the four TU datasets, with positive lower confidence bounds (Figure 3b), so the benefit cannot be explained solely by giving Mixed more channels (complete scans in Appendix J).

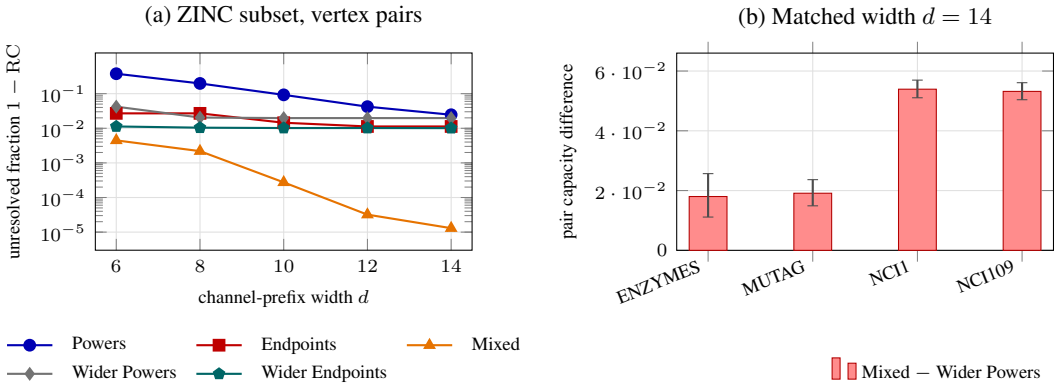


Figure 3: Object discrimination depends on the object domain as well as on width. (a) On ZINC subset pairs, endpoints outperform powers at every width (log scale of $1 - \mathrm{RC}$). (b) At matched width $d = 14$, Mixed exceeds Wider Powers on all four TU pair domains (95% intervals).

Node and edge gains are smaller and can reverse. The ZINC subset gains are 1.5×10^{-5} on nodes and 3.9×10^{-5} on edges, and the QM9 gains are 1.8×10^{-4} and 4.4×10^{-4}; only the ZINC intervals include zero. Mixed has lower capacity than Wider Powers on NCI1, NCI109, and molhiv nodes and edges and on ENZYMES and PROTEINS edges, with intervals strictly below zero.

High average capacity is not complete resolution. On ZINC subset pairs at $d = 14$, 10,735 of 12,000 graphs retain unresolved comparisons under Powers and 23 under Mixed, so an average RC near one does not imply full orbit recovery.

6.3 Scope of the Downstream Comparison

The measurements above concern structural discrimination, not prediction. Downstream prediction is discussed only in the published GNNPlus setting (Luo et al., 2025), which this paper does not rerun or score (Appendix K.2): a pair-domain gain cannot be transferred to a model that receives encodings only on nodes or existing edges, and even on a matching domain RC measures neither target relevance nor what training can learn.

7 Discussion and Conclusion

Structural discrimination depends jointly on encoding operations, finite budgets, readout, and object domain: the same degree budget can imply widths that differ by orders of magnitude, channelwise powers are injective on exact multisets yet change what a mean readout observes, and joint triple first moments collapse on equal-size regular graphs with minimum degree at least one. Empirically, pair-domain gains coexist with small or negative node and edge changes.

Three limitations bound these conclusions: the membership bounds are sufficient rather than tight, so the finite dictionaries need not recover full FWL; the object comparisons depend on numerical tolerances and eligibility, so a reported mean describes the eligible graphs; and the archived BREC and PPGN records are outputs of the described implementations. Closing the gap between sufficient membership bounds and strict round separation would require inputs that every seed-input network in a specified class maps together while an explicit-input network separates them.

Reproducibility Statement

The constructions and protocols needed to reproduce the reported measurements are specified in the referenced sections: encoding definitions and budgets in Section 3, the BREC and PPGN protocols and their archived counts in Appendices E.1 and G, and the within-graph protocol and results in Appendices E.3 and K. Dataset sources are in Table 3, and code is at the anonymous repository in the scope statement.

AI USAGE DISCLOSURE

We used generative AI tools to assist in drafting sections of this paper. The authors take responsibility for the final content of the paper, including the accuracy and originality of all text and claims produced with AI assistance.

REFERENCES

Louis Airale, Antonio Longa, Mattia Rigon, Andrea Passerini, and Roberto Passerone. Simple path structural encoding for graph transformers. In *International Conference on Machine Learning*, pp. 857–873, 2025.

Linus Bao, Emily Jin, Michael Bronstein, İsmail İlkan Ceylan, and Matthias Lanzinger. Homomorphism counts as structural encodings for graph learning. In *International Conference on Learning Representations*, 2025.

Mitchell Black, Zhengchao Wan, Gal Mishne, Amir Nayyeri, and Yusu Wang. Comparing graph transformers via positional encodings. In *International Conference on Machine Learning*, pp. 4103–4139, 2024.

Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, 2005. doi: 10.1093/bioinformatics/bti1007.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:657–668, 2023. doi: 10.1109/TPAMI.2022.3154319.

Rongqin Chen, Shenghui Zhang, Leong Hou U, and Ye Li. Redundancy-free message passing for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Rongqin Chen, Yan Li, Dan Wu, Fan Mo, Shenghui Zhang, Pak Lon Ip, Hoi Cheong Iam, Ye Li, and Leong Hou U. Enhanced subgraph learning in 2-FWL GNNs via local connectivity, spectral, and distance encodings. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 227–238, 2025. doi: 10.1145/3711896.3736933.

Rongqin Chen, Fan Mo, Pak Lon Ip, Shenghui Zhang, Dan Wu, Ye Li, and Leong Hou U. Connectivity-Guided Sparsification of 2-FWL GNNs: Preserving Full Expressivity with Improved Efficiency. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):20226–20234, 2026. doi: 10.1609/aaai.v40i24.39110.

A. K. Debnath, R. L. Lopez de Compadre, G. Debnath, A. J. Shusterman, and C. Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786–797, 1991. doi: 10.1021/jm00106a046.

Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003. doi: 10.1016/S0022-2836(03)00628-4.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations*, 2022a.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22326–22340, 2022b.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Jiarui Feng, Lecheng Kong, Hao Liu, Dacheng Tao, Fuhai Li, Muhan Zhang, and Yixin Chen. Extending the design space of graph neural networks by rethinking folklore Weisfeiler-Lehman. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Floris Geerts and Juan L. Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022.

Florian Grötschla, Jiaqing Xie, and Roger Wattenhofer. Benchmarking positional encodings for GNNs and graph transformers. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 2663–2674, 2026. doi: 10.1145/3770854.3785701.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

John J. Irwin, Teague Sterling, Michael M. Mysinger, Erin S. Bolstad, and Ryan G. Coleman. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52 (7):1757–1768, 2012. doi: 10.1021/ci3001277.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Derek Lim, Joshua Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and basis invariant networks for spectral graph representation learning. In *International Conference on Learning Representations*, 2023.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Can classic GNNs be strong baselines for graph-level tasks? simple architectures meet excellence. In *International Conference on Machine Learning*, pp. 41290–41310, 2025.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, pp. 23321–23337, 2023.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014. doi: 10.1016/j.jsc.2013.09.003.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4602–4609, 2019.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+)*, 2020.

Pál András Papp and Roger Wattenhofer. A theoretical comparison of graph neural network extensions. In *International Conference on Machine Learning*, pp. 17323–17345, 2022.

Jason Piquenot, Aldo Moscatelli, Maxime Bérar, Pierre Héroux, Jean-Yves Ramel, Romain Raveaux, and Sébastien Adam. G2N2: Weisfeiler and Lehman go grammatical. In *International Conference on Learning Representations*, 2024.

Omri Puny, Derek Lim, Bobak Kiani, Haggai Maron, and Yaron Lipman. Equivariant polynomials for graph neural networks. In *International Conference on Machine Learning*, pp. 28191–28222, 2023.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014. doi: 10.1038/sdata.2014.22.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. BRENDA, the enzyme database: Updates and major new developments. *Nucleic Acids Research*, 32(Database issue):D431–D433, 2004. doi: 10.1093/nar/gkh081.

Nikil Wale and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. In *Sixth IEEE International Conference on Data Mining*, pp. 678–689, 2006. doi: 10.1109/ICDM.2006.39.

Yanbo Wang and Muhan Zhang. An empirical study of realized GNN expressiveness. In *International Conference on Machine Learning*, pp. 52134–52155, 2024.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/C7SC02664A.

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

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Bohang Zhang, Guhao Feng, Yiheng Du, Di He, and Liwei Wang. A complete expressiveness hierarchy for subgraph GNNs via subgraph Weisfeiler-Lehman tests. In *International Conference on Machine Learning*, pp. 41019–41077, 2023.

A Proof of the Algebraic Characterization

We prove Theorem 1 on a fixed finite graph. Recall that W_ℓ is the real span of the round-ℓ color indicators, and $H_{\ell+1}$ is the polynomial closure of H_ℓ and its contractions.

Base case and joint counts. By definition, $H_0 = W_0$. Suppose $H_\ell = W_\ell$ and write $e_c(\mathbf{v}) = \mathbf{1}\{C_\ell(\mathbf{v}) = c\}$ for each attained color c. For a color pattern $\boldsymbol{c} = (c_1, \ldots, c_k)$, define

$$N_{\boldsymbol{c}}(\mathbf{v}) = \mathrm{Contract}(e_{c_1}, \ldots, e_{c_k})(\mathbf{v}).$$

This counts pivots whose replacement tuples have the joint color pattern $\boldsymbol{c}$. The old-color indicators and all these counts form a finite profile that determines the next color in equation 1.

Every generated channel respects the next coloring. Expand each input $T_i \in H_\ell$ in the basis $(e_c)_c$. Multilinearity expresses $\mathrm{Contract}(T_1, \ldots, T_k)$ as a linear combination of the channels $N_{\boldsymbol{c}}$. Each such count, and each retained old-color indicator, is constant on a next-round color class. Finite linear combinations and Hadamard products preserve this property. Thus $H_{\ell+1} \subseteq W_{\ell+1}$.

Every next-round color indicator is generated. List the old-color indicators and joint counts as $f_1, \ldots, f_s$. For one channel f let A_f be its finite set of attained values. For $a \in A_f$, the polynomial

$$p_{f,a}(f) = \prod_{\substack{b \in A_f \\ b \neq a}} \frac{f - b}{a - b}$$

is the indicator of $f = a$. Its denominators are nonzero; an empty product equals the constant-one channel. It therefore belongs to $H_{\ell+1}$. For each attained profile $(a_1, \ldots, a_s)$, the product

$$\prod_{j=1}^{s} p_{f_j, a_j}(f_j)$$

indicates exactly one next-round color class. Hence $W_{\ell+1} \subseteq H_{\ell+1}$. Induction proves the theorem. □

What interpolation does and does not establish. Interpolation recovers the partition, not necessarily the numerical labels assigned by HASH. It proves representability by the allowed algebraic operations; its degree can depend on the graph, and preprocessing need not evaluate it. Hashing remains a valid implementation of refinement and can also encode restricted profiles. Our fixed numerical features avoid lookup-based assignment of shared profile IDs, but their bounded channelwise powers need not separate all profiles. Neither polynomial representability nor differentiability alone establishes implementation by a given backbone.

Comparing a finite family of graphs. The same argument applies to the disjoint union of their tuple domains, with each contraction summing only over vertices of the tuple's graph. Use shared initial colors and interpolate over the union of attained profiles at each round. The coefficients then agree across the family. Likewise, degree normalization can interpolate over the union of degree values. This gives the stated FWL upper bound when comparing any fixed finite collection of graphs; it does not give a bounded dictionary that is complete uniformly over all graph sizes.

B Examples of Pair and Triple Features

These examples show what the exact operations compute before imposing a dictionary budget. Throughout, the graph is simple and undirected, d_u is the degree of u, and H_ℓ has the tuple order of the current subsection. Algebraic membership and evaluation cost are separate questions.

B.1 Pair Features

For $k = 2$, channels are matrices. The replacement convention gives

$$\mathrm{Contract}(T_1, T_2)(u, v) = \sum_t T_1(t, v) T_2(u, t) = (T_2 T_1)_{uv}.$$

This identity holds for arbitrary pair matrices: setting $T_1 = Y$ and $T_2 = X$ gives $\mathrm{Contract}(Y, X) = XY$. Symmetry alone does not imply that two matrices commute; polynomial seeds commute

because they are polynomials in the same matrix M. Thus $X, Y \in H_\ell$ imply $XY \in H_{\ell+1}$. For color-indicator matrices, $(XY)_{uv}$ counts pivots for which (u,t) and (t,v) have the specified colors.

First-round features. Since $I, A, J \in H_0$, the following channels all belong to H_1.

Channel	Meaning of its (u,v) entry
A^2	Number of common neighbors
$AJ,\ JA$	Endpoint degrees $d_u,\ d_v$
$I \odot A^2$	d_u on the diagonal; zero elsewhere
$A \odot A^2$	Number of triangles containing edge (u,v)
$(J - I - A) \odot A^2$	Common-neighbor count on distinct nonedges
$(AJ) \odot (JA)$	Product $d_u d_v$
$A^2 \odot A^2$	Squared common-neighbor count

Degree vectors cost $O(n + |E|)$ to compute. Dense matrix multiplication computes A^2 in $O(n^3)$ operations; sparse multiplication may reduce this cost. Given the required matrices, each listed Hadamard operation costs $O(n^2)$ for dense output.

Nonlinear tests versus further contractions. Hadamard products and polynomial tests of newly available channels remain in the same $H_{\ell+1}$. For example, indicators of attained common-neighbor counts belong to H_1 by Appendix A. Their interpolation degrees need not be small. By contrast, multiplying newly generated matrices uses another contraction: $X, Y, Z \in H_\ell$ guarantees $(XY)Z \in H_{\ell+2}$, without asserting that it is absent from $H_{\ell+1}$. Appendix C.2 makes the sufficient round bounds precise.

B.2 Triple Features

For $k = 3$, a channel assigns a scalar to (u,v,w), and

$$\mathrm{Contract}(T_1, T_2, T_3)(u,v,w) = \sum_t T_1(t,v,w) T_2(u,t,w) T_3(u,v,t).$$

Write $X_{12}(u,v,w) = X_{uv}$, $X_{13}(u,v,w) = X_{uw}$, and $X_{23}(u,v,w) = X_{vw}$. The lifts of I, A, and the constant-one channel $\mathbf{1}_3$, belong to H_0.

What a common pivot adds. The triple common-neighbor count $c(u,v,w) = \sum_t A_{ut} A_{vt} A_{wt}$ requires the same vertex to be adjacent to all three endpoints. It is a joint statistic, not a product of separate pair counts. The following contractions belong to H_1:

Contraction	Value at (u,v,w)
$\mathrm{Contract}(\mathbf{1}_3, A_{12}, \mathbf{1}_3)$	d_u
$\mathrm{Contract}(A_{12}, A_{12}, \mathbf{1}_3)$	$(A^2)_{uv}$
$\mathrm{Contract}(A_{13}, A_{12}, \mathbf{1}_3)$	$(A^2)_{uw}$
$\mathrm{Contract}(A_{13}, A_{12}, A_{23})$	$c(u,v,w)$

Coordinate permutations give the other endpoint and pair variants. Hadamard combinations such as $d_u d_v d_w$ and $(A^2)_{uv}(A^2)_{uw}$ remain in H_1. The feature $A_{uv}A_{uw}A_{vw}c(u,v,w)$ counts vertices extending the specified triangle to a four-clique. This is an interpretation of an algebraic channel, not an additional encoding baseline.

Other joint adjacency patterns. For $\boldsymbol{\varepsilon} \in \{0,1\}^3$, let $F_1 = A$ and $F_0 = J - A$. Then

$$\mathrm{Contract}\big((F_{\varepsilon_3})_{13}, (F_{\varepsilon_1})_{12}, (F_{\varepsilon_2})_{23}\big)(u,v,w) = \sum_t (F_{\varepsilon_1})_{ut} (F_{\varepsilon_2})_{vt} (F_{\varepsilon_3})_{wt}.$$

These eight channels count joint adjacency patterns. Since $J - A$ includes diagonal entries, a nonadjacent pivot may coincide with an endpoint. Use $J - I - A$ instead when it must also be distinct from that endpoint.

Connection to the proposed features. Replacing adjacency factors by the polynomial channels gives

$$K_{(a,b,c)}(u,v,w) = \mathrm{Contract}\big((S_c)_{13}, (S_a)_{12}, (S_b)_{23}\big)(u,v,w).$$

This is the triple case of equation 4. For one triple, $c(u, v, w)$ can be computed by intersecting neighbor sets. Once polynomial channels are available, a direct evaluation of $K_{(a,b,c)}$ costs $O(n)$ per triple, or $O(Rn)$ for R requested triples. Dense output costs $O(n^4)$ operations and $O(n^3)$ storage per channel. Accumulating over pivots avoids an n^4 intermediate tensor. Selected evaluations can be inexpensive even when dense tuple encoding is not.

C Guarantees for the Budgeted Encodings

We first prove the round bounds for every $k \geq 2$, then show that the numerical construction commutes with vertex relabeling. The round bounds are statements on a fixed finite graph, not graph-independent interpolation recipes or neural-network depth lower bounds. We then specify channel ordering, dictionary sizes, and evaluation costs.

C.1 Formal Encoding Definition

Fix $k \geq 2$ and budgets $Q, P \geq 1$. The following dictionaries are indexed lists. Both bases start with $S_0 = I$ and $S_1 = M$. For monomials, $S_{q+1} = MS_q$; for Chebyshev channels, $S_{q+1} = 2MS_q - S_{q-1}$. Ordering and duplicate conventions are specified in Appendix C.6. For $k = 2$, define

$$\mathcal{S} = (I, J, A, S_1, \ldots, S_Q), \qquad \mathcal{D} = \mathcal{S} \cup \{B_{a,b} : a, b \geq 0,\ 0 < a + b < Q\}.$$

Writing $h_a = S_a\mathbf{1}$, use $J = \mathbf{1}\mathbf{1}^\top$ and the symmetry of S_b to obtain

$$B_{a,b}(u, v) = h_a(u)h_b(v), \qquad B_{a,b} = (S_aJ) \odot (JS_b).$$

The corresponding three-matrix product satisfies

$$S_aJS_b = (S_a\mathbf{1})(\mathbf{1}^\top S_b) = (S_a\mathbf{1})(S_b\mathbf{1})^\top = h_a h_b^\top.$$

Here $\mathbf{1}^\top S_b = (S_b\mathbf{1})^\top$ uses symmetry; for a general matrix its column sums need not equal its row sums. The Hadamard product above and the triple product both give $B_{a,b}$. Ordinary multiplication of the two factors instead gives $(S_aJ)(JS_b) = S_aJ^2S_b = nB_{a,b}$. The implementation evaluates the two vectors and their outer product. For $k \geq 3$, write $X_{ij}(\mathbf{v}) = X(v_i, v_j)$ for a pair-matrix lift and $\mathbf{1}_k$ for the constant-one tuple channel. Replace the pair dictionaries by

$$\begin{aligned}\mathcal{S} &= \{\mathbf{1}_k\} \cup \{X_{ij} : X \in \{I, A, S_1, \ldots, S_Q\},\ i \neq j\},\\ \mathcal{D} &= \mathcal{S} \cup \{(B_{a,b})_{ij} : i \neq j,\ a, b \geq 0,\ 0 < a + b < Q\}\\ &\quad \cup \{K_{\mathbf{q}} : \mathbf{q} \in \{0, \ldots, Q\}^k\}.\end{aligned}$$

The lifted equality and adjacency channels determine the atomic type of an unlabeled tuple. Labeled graphs may require additional atomic-type indicators; the dictionary counts below concern the unlabeled construction. For the appropriate tuple order, the two encodings are

$$E_{\text{seed}}(\mathbf{v}) = (T(\mathbf{v})^p)_{\substack{T \in \mathcal{S} \\ 1 \leq p \leq P}}, \qquad E_{\text{explicit}}(\mathbf{v}) = (T(\mathbf{v})^p)_{\substack{T \in \mathcal{D} \\ 1 \leq p \leq P}}. \tag{8}$$

Thus the power expansion acts separately on each selected channel; it does not append further mixed products.

C.2 Round Bounds

Proposition 1 (round bounds)**.** *On each fixed finite graph, for either polynomial basis and every $k \geq 2$, set $L = 1 + \lceil \log_2 Q \rceil$. Seed channels belong to H_L and explicit channels to H_{L+1}, for every finite $P \geq 1$. These are membership bounds, not completeness or optimality guarantees.*

Proof. **Normalization is available in the first round.** For pairs, AJ and JA give both endpoint degrees in H_1. For $k \geq 3$, contract A_{ij} in slot j and use constant-one channels in all other slots; this gives the degree of coordinate i. The set $\{0\} \cup \{d_v : v \in V\}$ is finite, so polynomial interpolation gives a scalar polynomial g with $g(0) = 0$ and $g(d) = d^{-1/2}$ at every positive value in this set. Polynomial evaluation stays in H_1, as does the pointwise product $M_{uv} = g(d_u)A_{uv}g(d_v)$. Thus M,

or each ordered pair lift M_{ij}, lies in H_1. The interpolating polynomial is chosen for the degree values of the fixed graph. In practice, normalization is computed directly from degrees.

One contraction multiplies pair matrices. For $k = 2$, $\mathrm{Contract}(Y, X) = XY$. For $k \geq 3$, put X_{ij} in slot j, Y_{ij} in slot i, and ones elsewhere. The result is $\sum_t X(v_i, t)Y(t, v_j) = (XY)_{ij}(\mathbf{v})$. Balanced products thus place M^q, or all its lifts, in $H_{1+\lceil \log_2 q \rceil}$ for $q \geq 1$. Each S_q is a linear combination of $I, M, \ldots, M^q$, so it obeys the same bound. Since I, A, J and their lifts lie in H_0, all seed channels lie in H_L. The logarithmic bound counts balanced matrix-multiplication stages. Hadamard products are already part of Alg and add no refinement rounds; in general, $M \odot M$ is different from M^2.

Endpoint and joint channels need at most one more round. The products $S_a J$ and $J S_b$ are available in H_{L+1} for $a, b \geq 0$ and $a + b < Q$. Their Hadamard product is $B_{a,b}$ and stays in the same space, including after lifting. For $k \geq 3$, use the cyclic lifts from equation 5: let $j(i)$ be the cyclic successor of i and set

$$T_i(\mathbf{x}) = S_{q_{j(i)}}(x_{j(i)}, x_i).$$

Each T_i is a seed lift in H_L. Since $j(i) \neq i$, replacing coordinate i by t supplies $S_{q_{j(i)}}(v_{j(i)}, t)$. Reindexing the product by the permutation j gives $\mathrm{Contract}(T_1, \ldots, T_k) = K_{\mathbf{q}} \in H_{L+1}$. Finally, Hadamard powers stay in the same algebra for every finite P. This proves both bounds. □

These are sufficient bounds, not strict round separations. For example, M^2 and MJM lie in H_2, as does $K_{(1,1,1)}$ in the order-three space. The corresponding unnormalized triple common-neighbor count already lies in H_1.

C.3 Equivariance

Let Π represent a permutation π and let $A' = \Pi A \Pi^\top$. Degrees permute in the same way, including at isolated vertices, so $M' = \Pi M \Pi^\top$. Also $\Pi I \Pi^\top = I$ and $\Pi J \Pi^\top = J$. For any pair matrices,

$$(\Pi X \Pi^\top)(\Pi Y \Pi^\top) = \Pi X Y \Pi^\top, \qquad (\Pi X \Pi^\top) \odot (\Pi Y \Pi^\top) = \Pi (X \odot Y) \Pi^\top.$$

Polynomial coefficients are fixed independently of the graph, so $S'_q = \Pi S_q \Pi^\top$. Thus every pair channel, and each coordinate lift, is equivariant. For $K_{\mathbf{q}}$, reindexing the pivot as $t' = \pi(t)$ gives

$$K_{\mathbf{q}}(G'; \pi(v_1), \ldots, \pi(v_k)) = K_{\mathbf{q}}(G; v_1, \ldots, v_k).$$

Hadamard powers preserve this identity, and the channel order is fixed independently of the graph. This proves equivariance of both encodings. Diagonal and existing-edge restrictions also commute with relabeling. Other tuple restrictions or aggregations must satisfy that same condition; an invariant readout then yields a graph invariant. These identities concern exact arithmetic. Numerical implementations require a specified floating-point tolerance.

C.4 What Changing the Polynomial Basis Preserves

Fix the same degree budget Q and retain every degree from zero to Q. Monomials and Chebyshev polynomials are two bases of this polynomial space: there is an invertible, graph-independent triangular matrix C such that $S'_a = \sum_b C_{ab} S_b$. Consequently,

$$B'_{a,b} = \sum_{c,d} C_{ac} C_{bd} B_{c,d}, \qquad K'_{\mathbf{q}} = \sum_{\mathbf{r}} \left(\prod_i C_{q_i r_i} \right) K_{\mathbf{r}}.$$

The inverse change of basis gives the reverse transformations. The omitted endpoint entry $B_{0,0} = J$ remains available in the seed block. Thus the complete seed and explicit dictionaries in either basis determine each other through fixed linear maps. This holds across graphs as well as within one graph, and preserves equality of tuple features and their multisets under exact arithmetic. It does not claim that restricted readouts or trained networks behave identically. First-moment pooling is a special case: averaging commutes with the fixed linear change of basis, so equality of the full mean vectors is also preserved on the same retained domain. Higher channelwise moments can mix into cross-channel products under a basis change; a readout retaining only marginal moments need not preserve this equivalence. After channelwise powers are appended, the first-power block still determines the other basis and all its powers; the transformation need not remain linear. Degree subsampling, compression, rounding, or a different underlying matrix can invalidate this argument.

C.5 What the Budgets Can Add

For $P \geq 1$, the map $x \mapsto (x_i^p)_{i,\,1\leq p\leq P}$ retains x and is determined by x. It neither creates nor removes distinctions between raw feature vectors, or between their exact multisets. It can change the statistics available to a restricted readout, such as sum pooling followed by a linear map. Such gains are readout-dependent, not a stronger raw encoding.

For a fixed polynomial basis, increasing Q retains the old dictionary and adds indexed channels; replacing seed by explicit also retains every old channel. These changes cannot reduce raw discrimination, but need not increase it strictly. A fixed-width selection or compression need not be nested. No claim is made that increasing these budgets recovers the entire k-FWL refinement algebra.

C.6 Dictionary Size and Evaluation Cost

Ordering and channel counts. We treat dictionaries as ordered lists of indexed channels: identical values do not remove an entry. Pair seeds follow the displayed order $(I, J, A, S_1, \ldots, S_Q)$; endpoint channels follow lexicographic order of (a, b), omitting $(0, 0)$ because $B_{0,0} = J$. For $k \geq 3$, the constant channel comes first, followed by seed lifts in seed order and lexicographic ordered-coordinate order (i, j). Endpoint lifts are ordered lexicographically by (i, j, a, b) and joint channels by $\mathbf{q}$. Within each channel, powers are ordered from 1 to P. These conventions are independent of the graph.

Let $e_Q = Q(Q+1)/2 - 1$ be the number of retained endpoint channels, and let σ_k and δ_k be the seed and explicit dictionary sizes. We use Greek letters here to keep these counts distinct from the retained object count $m = |\Omega(G)|$ of Section 3. For pairs,

$$\sigma_2 = Q + 3, \qquad \delta_2 = \sigma_2 + e_Q.$$

For $k \geq 3$, every nonconstant pair seed is lifted to $k(k-1)$ ordered coordinate pairs. Hence

$$\sigma_k = 1 + k(k-1)(\sigma_2 - 1),$$
$$\delta_k = \sigma_k + k(k-1)e_Q + (Q+1)^k.$$

The encodings have $\sigma_k P$ and $\delta_k P$ entries per retained tuple. These count indexed features, not linearly independent channels. For example, symmetry duplicates some pair lifts; on a positive-degree regular graph, every $B_{a,b}$ equals J.

Pair preprocessing. Let R be the number of retained positions: n for node diagonals, the number of directed edge entries for existing edges, or n^2 for all pairs. We write R rather than r to avoid a clash with the orbit count of Section 6. The following are direct evaluation bounds:

Computation	Cost
Dense polynomial sequence $S_1, \ldots, S_Q$	$O(Qn^3)$ operations
Sparse propagation of $h_0, \ldots, h_Q$	$O(Q(n + \lvert E\rvert))$ operations
Endpoint-channel entries	$O(Re_Q)$ operations
Hadamard expansion	$O(R\delta_2 P)$ operations
Explicit encoding storage	$O(R\delta_2 P)$ scalars

Endpoint propagation starts with $h_0 = \mathbf{1}$ and $h_1 = M\mathbf{1}$. For monomials, $h_{a+1} = Mh_a$; for Chebyshev channels, $h_{a+1} = 2Mh_a - h_{a-1}$. It computes vectors, not the matrix seeds. Materializing $B_{a,b}(u, v) = h_a(u)h_b(v)$ incurs the separate output cost above. Restricting output positions saves storage but does not itself avoid computing the seeds. Keeping all dense polynomial channels requires $O(Qn^2)$ working storage, in addition to the output; endpoint vectors require $O(Qn)$ storage. Streaming can trade retained intermediates for recomputation.

Higher-order evaluation. For $k \geq 3$ and R requested tuples, pair lifts reuse the same matrices and endpoint vectors. Evaluating all endpoint lifts directly costs $O(Rk(k-1)e_Q)$ operations. Once the polynomial channels are available, direct evaluation of all $K_{\mathbf{q}}$ costs $O(Rnk(Q+1)^k)$ operations. Accumulating over pivots avoids storing a pivot-by-tuple tensor. Hadamard expansion and explicit output storage cost $O(R\delta_k P)$. Dense evaluation sets $R = n^k$, so both tuple count and joint-channel count limit feasible k and Q. These are implementation upper bounds, not lower bounds on every possible algorithm.

D Backbone Reconstruction and Its Limits

This appendix supports the seed/explicit distinction. Reconstruction depends on both the backbone's operators and its access to the pair or tuple states on which they act. A generic tensor product is not sufficient: the required index contractions and pointwise products must be implemented.

D.1 A Sufficient Reconstruction Criterion

For a backbone class B and encoding E, let $\mathrm{Rep}(B, E)$ denote its function class with unrestricted finite depth and width and a fixed output head.

Proposition 2 (synthesis and simulation)**.** *If B can exactly construct the explicit channels from the seeds, retain the required intermediate states, compose this construction with its networks, and select the seed channels from explicit inputs, then*

$$\mathrm{Rep}(B, E_{\mathrm{seed}}) = \mathrm{Rep}(B, E_{\mathrm{explicit}}).$$

Proof. A seed-input network can prefix the construction to an explicit-input network. Conversely, an explicit-input network can discard the extra channels. These two simulations prove the equality. □

The result concerns unrestricted function classes. At fixed depth or width, precomputation may still save computation or improve optimization. We next give sufficient circuits for the dictionaries in Section 3. They start with the supplied polynomial seeds; normalization need not be learned or computed by interpolation.

D.2 Pair Inputs

Suppose the backbone maintains all ordered pairs and supports exact matrix multiplication, exact channelwise multiplication, and channel retention and routing. The $p = 1$ seed block already contains $I, J, A, S_1, \ldots, S_Q$. In parallel, form

$$U_a = S_a J, \qquad V_b = J S_b \qquad (0 \le a, b < Q),$$

then form $B_{a,b} = U_a \odot V_b$ only for $0 < a + b < Q$. These are exactly the extra pair dictionary entries. Balanced channelwise multiplication trees produce all powers $B_{a,b}^{\odot p}$, $1 \le p \le P$; seed powers are already supplied. If a layer can perform arbitrarily many copies of one permitted primitive in parallel, this circuit has depth at most $2 + \lceil \log_2 P \rceil$, plus routing layers. The first stage uses matrix multiplication, the second uses Hadamard multiplication. Width must accommodate their intermediate channels. This is a sufficient algebraic circuit, not a claim about every PPGN implementation.

D.3 Arbitrary Tuple Inputs

Suppose instead that the backbone maintains all ordered k-tuples and supports exact $\mathrm{Contract}$, channelwise multiplication, and retention and routing. The seeds contain every ordered lift of S_q and the constant channel. For $i \neq j$, contracting $(S_a)_{ij}$ in slot j and ones elsewhere gives $h_a(v_i)$. Contracting $(S_b)_{ij}$ in slot i gives $h_b(v_j)$, using symmetry of S_b. Their pointwise product gives $(B_{a,b})_{ij}$. The cyclic-slot construction in Appendix C.2 computes each $K_{\mathbf{q}}$ in the first contraction stage as well. After the endpoint-product stage, balanced multiplication trees supply all requested powers. Under the same parallel-layer convention, the depth is again at most $2 + \lceil \log_2 P \rceil$, plus routing. For fixed k, Q, P the width is finite, but grows with the dictionary, including its $(Q+1)^k$ joint channels.

D.4 Implications for Different Backbones

Tensor-based GNNs. The displayed circuits apply only when the layer primitives implement the specified contractions, pointwise products, retention, and routing. One shared-pivot contraction produces the joint channels, but endpoint products and the power expansion may require further stages. The stated circuit depth is therefore not a universal neural-layer count. Approximate multiplication does not establish equality of exact function classes, and algebraic membership alone is not an implementation proof.

MPNNs. After restricting inputs to diagonals or existing edges, the full pair grid used by the circuits is no longer directly available. Computing explicit channels before this restriction can expose aggregated statistics at retained positions. Failure of the displayed circuit, however, does not prove that a particular MPNN cannot reconstruct those statistics by another route; that requires an architecture-specific argument.

Graph transformers. Access to all pairs is not sufficient by itself: global attention should not be identified with the particular products and pivot sums used here. Some architectures may implement these operations, while others do not provide them directly. Conclusions must specify the maintained states, attention and update operators, and input scheme rather than rely on the model name.

What would establish a strict separation? Pointwise powers alone do not suffice: a sufficiently expressive local encoder can approximate them on compact input domains. A strict separation requires inputs indistinguishable by every seed-input network in the specified class but distinguishable by an explicit-input network. Neither the absence of our reconstruction circuit nor a downstream performance gain supplies that proof. Finite-width training outcomes measure performance under the chosen architecture and resource budget.

D.5 Scope of the Reconstruction Argument

The circuits establish existence under explicit algebraic primitives. They do not show that a trained network learns those circuits, or that precomputation has no practical benefit at a fixed depth and width. The neural BREC records in Appendix G are not tests of equality of unrestricted function classes.

E Encoding Evaluation Protocols

The fixed-encoding studies use simple, unweighted, undirected topology. Node and edge attributes are excluded. The two campaigns have different precision and selection conventions and are reported separately.

E.1 Fixed-Encoding BREC Protocol

Reference groups. The 400 official pair IDs partition as follows under the deterministic joint-replacement reference:

Reference	Pair IDs	Count
2-FWL separates	0–109, 160–319	270
Only-3-FWL separates	110–159, 320–339, 360–399	110
Neither separates	340–359	20

These are reference labels, not outcomes inferred from an encoding. For deterministic non-GNN methods, BREC does not require the RPC procedure (Wang & Zhang, 2024, Appendix O). Our feature construction is permutation equivariant and its multiset and aggregate readouts are permutation invariant in exact arithmetic. Separately, every channel of the order-2 and order-3 dictionaries lies in H_L with $L = 1 + \lceil \log_2 Q \rceil$ (Proposition 1), so no channel can separate tuples that round-L 2-FWL or 3-FWL leaves in one class; this upper bound excludes pairs 340–359 for both orders. We therefore omit their computation in the fixed-encoding campaign and record them as theoretical exclusions, not measured failures. Benchmark coverage retains the denominator 400; empirical positive-cohort coverage uses 380, subject to any explicitly stated resource exclusions. The order-2 conditions additionally exclude the 110 only-3-FWL pairs.

Features and readouts. The archived implementation uses $M = D^{-1/2}AD^{-1/2}$ and float32, with constant, identity, adjacency and polynomial channels, six ordered pair lifts for triples, and the endpoint/joint augmentations. Joint channels are normalized by n in the primary comparison. The maximum $Q = 16$, $P = 4$ pass supplies slices for $Q = 2, 4, 8, 16$ and $P = 1, 2, 4$. Multisets compare SHA256 digests of tuple rows rounded to eight decimals. Means aggregate channelwise powers over all ordered tuples, divide by tuple count, round, and compare with absolute tolerance

10^{-8}. Only the first stored A/B instances are evaluated; no reliability block or random permutation is computed. Numerical results are not exact arithmetic equality proofs.

Coverage and provenance. The supplied report accounts for all 400 IDs in every configuration: 270 measured plus 130 theoretical exclusions for pairs, and 380 measured plus 20 exclusions for triples. It contains no execution failures or incomplete conditions. Invariance checks are explicitly unperformed; zero flagged records must not be interpreted as passed checks. Counts and source provenance are detailed in Appendix F. The supplied archive includes frozen encoding code, a run plan, and producer manifests, which establish the recorded implementation identity.

Integrated scale comparison. At joint $Q = 4$, both graphs also supply moments of K, K/n, and moments restored to K units. The same report records rounded and unrounded decisions (Appendix F.1). The current runner uses absolute tolerance 10^{-8} and, for unrounded comparisons, relative tolerance 10^{-10}. These are tolerance-based numerical comparisons; no separate relabeling experiment is required.

E.2 Dataset Sources and Graph Versions

Table 3 distinguishes original data sources from the graph benchmark versions used in this study. MUTAG, NCI1, NCI109, ENZYMES, and PROTEINS are distributed through TUDataset (Morris et al., 2020). The ZINC and QM9 reruns use PyTorch Geometric loaders (Fey & Lenssen, 2019); their reported graph counts describe the processed collections, rather than the sizes of the original databases. The short names molhiv, pcba, and PPA refer to OGB's `ogbg-molhiv`, `ogbg-molpcba`, and `ogbg-ppa`. VOC denotes PascalVOC-SP, and Peptides-functional denotes Peptides-func in LRGB. Our topology-only comparison protocol is specified below; it does not use the prediction targets or constitute a rerun of the original benchmark's supervised evaluation.

Table 3: Sources of the evaluated graph datasets. Original-data references and graph-benchmark references describe different stages of provenance.

Dataset	Data source and graph benchmark
ZINC subset / full	ZINC molecular database (Irwin et al., 2012); graph benchmark and subset protocol (Dwivedi et al., 2023).
QM9	Quantum-chemistry molecular dataset (Ramakrishnan et al., 2014).
MUTAG	Mutagenicity data (Debnath et al., 1991); TU graph collection (Morris et al., 2020).
NCI1 / NCI109	Chemical-compound benchmarks (Wale & Karypis, 2006); TU graph collection (Morris et al., 2020).
ENZYMES	BRENDA enzyme data (Schomburg et al., 2004) and protein graph representation (Borgwardt et al., 2005).
PROTEINS	Protein classification data (Dobson & Doig, 2003) and graph representation (Borgwardt et al., 2005).
MNIST	Handwritten-digit images (LeCun et al., 1998); superpixel graph benchmark (Dwivedi et al., 2023).
CIFAR10	Image collection (Krizhevsky, 2009); superpixel graph benchmark (Dwivedi et al., 2023).
CLUSTER / PATTERN	Stochastic-block-model graph benchmarks (Dwivedi et al., 2023).
PPA	Protein-association graph benchmark in OGB (Hu et al., 2020).
molhiv / pcba	MoleculeNet molecular tasks (Wu et al., 2018); OGB graph versions (Hu et al., 2020).
VOC	PASCAL VOC images (Everingham et al., 2010); PascalVOC-SP graph benchmark in LRGB (Dwivedi et al., 2022b).
Peptides-functional	Peptides-func graph benchmark in LRGB (Dwivedi et al., 2022b).

E.3 Within-Graph Object Discrimination

Data and objects. Dataset coverage is listed in Appendix K. Atom labels, bond labels, and graph-class labels are not encoding inputs. We evaluate nodes, unordered existing edges, and unordered pairs of distinct vertices separately. The protocol corresponds to `pair_resolution_representative_capacity_v13_float32`.

Width-controlled features. This experiment uses monomial channels of the symmetric-normalized operator $D^{-1/2}AD^{-1/2}$. Write $\mathcal{P}_d$ for the first d powers and $\mathcal{B}_d$ for the first d endpoint channels. Node powers start at degree two; edge and pair powers start at degree one. Endpoint indices satisfy $a, b \geq 0$, $a \leq b$, and $a + b \geq 1$, ordered by total degree and then a: $(0,1), (0,2), (1,1), (0,3), (1,2), \ldots$.

Route	Features	Channels per orientation
Powers	$\mathcal{P}_d$	d
Endpoints	$\mathcal{B}_d$	d
Mixed	$\mathcal{P}_d \oplus \mathcal{B}_d$	$2d$
Wider powers	$\mathcal{P}_{2d}$	$2d$
Wider endpoints	$\mathcal{B}_{2d}$	$2d$

We use $d = 6, 8, 10, 12, 14$. These are fixed channel prefixes, not the complete Q-bounded dictionaries of the BREC experiment; no additional Hadamard expansion or graph-level pooling is applied. Nodes use diagonal entries. Each unordered edge or pair stores both orientation vectors, so $a \leq b$ does not discard its transposed information. Width matching controls selected channels per orientation, not the number of independent scalars or computation time.

Object comparison. Two scalar values are close when

$$|x - y| \leq 10^{-8} + 10^{-6} \max(|x|, |y|).$$

Orientation vectors match only when every channel is close. Two unordered objects match if both orientations agree directly or both agree after one common swap. The orientation choice must be shared across all channels, including both branches of Mixed. This removes dependence on which endpoint was listed first. Nodes require just their single diagonal vector to match. Features use float32; aggregate statistics use float64. No column standardization is used in the reported v13 results.

Complementary metrics. For one graph and object domain, let $\mathcal{O}$ be its objects, r the number of automorphism orbits, T_{orb} the unordered object pairs in different orbits, and S the object pairs separated by the encoding. Form a similarity graph on $\mathcal{O}$ by joining matching objects, and let α be its maximum independent-set (MIS) size. Report

$$\text{Representative Capacity} = \frac{\alpha}{r}, \qquad \text{Pairwise Resolution} = \frac{|S \cap T_{\mathrm{orb}}|}{|T_{\mathrm{orb}}|}.$$

Also retain $\alpha/|\mathcal{O}|$ as the object-normalized representative fraction. The first metric asks how many mutually distinguishable representatives can be selected; the second asks how many eligible object pairs are separated. Two orbits of sizes a, b contribute ab pairs to the latter, so it is not an equally weighted average over orbit pairs. Neither metric alone certifies recovery of the entire orbit partition.

Automorphism orbits are computed from the topology with nauty (McKay & Piperno, 2014) via `pynauty`. Stable 2-FWL supplies a secondary reference: use $C(u, u)$ for nodes and the unordered colour pair $\{C(u, v), C(v, u)\}$ for edges and vertex pairs. Replacing T_{orb} by the unequal-colour pairs T_{WL} gives WL coverage. The ratio $|T_{\mathrm{WL}}|/|T_{\mathrm{orb}}|$ indicates how much of the orbit target the reference itself can separate. Full partition recovery requires both no missed cross-class pair and no separated same-class pair. Zero denominators are not applicable; empty domains are not counted as attaining a ceiling.

Endpoint comparisons and numerical checks. Compare Mixed with Powers to measure augmentation, and with Wider powers to compare endpoints with additional powers at the same channel count. Also compare the standalone families at d and $2d$. For each comparison, report the cross-orbit object pairs separated only by the first route, only by the second, by both, and by neither.

Check for same-orbit and same-WL separations, changes to the aligned separation mask under relabelling, and any separation lost when endpoints are appended. The stated run uses three independent permutations per graph. Actual check counts are recorded; zero checks are not a measured stability pass. A route's own scores use its own applicable checks. A paired comparison or baseline-dependent gain requires both routes to pass; an unreliable powers baseline does not invalidate Mixed's standalone score. The auxiliary fraction of powers collisions resolved by augmentation uses all baseline collisions, including same-orbit pairs. Its denominator and graph IDs are reported separately from orbit-normalized resolution.

Aggregation and computation. Keep per-graph values and report macro means, micro pair coverage, paired mean differences, wins/ties/losses, and full-partition recovery. Comparisons use common reliable graphs, with graph IDs and counts. The MIS search uses a budget of 200,000 expanded states per solve. A budget-limited feasible set is reported as a lower bound, not an exact Representative Capacity; exact capacity summaries require proved optimality on the numerical similarity graph. Representative indices are saved and their pairwise separation is checked. Graph-bootstrap intervals use 10,000 resamples of dataset records; repeated topologies retain their dataset multiplicity. Sign-flip p-values are exploratory, not the primary evidence of improvement. The optional equivalence flag uses a predeclared margin of 0.01 and requires the whole 95% interval to lie inside that margin. Outcome-selected subsets are descriptive only. If numerical or search exclusions occur, means describe the eligible graphs rather than the full dataset.

One graph is one task; widths share feature generation and prefix comparisons, and identical route widths reuse masks and MIS results. Stored masks are bit-packed and object-pair details are written out graph by graph. Worker threads and concurrent estimated memory are limited; this estimate is not a hard process-memory bound. Per-graph records, source/input identities, and costs accompany summaries. A flat width curve is an observed budget plateau, not proof of algebraic saturation or a downstream learning claim.

F Archived BREC Counts

The tables reproduce the monomial subset of `report-20260925-053512-1092765.json`. The report declares direct evaluation, no invariance checks, no adaptive skips, and no execution-error records. Each configuration accounts for all 400 pair IDs: 270 are measured for order 2 and 380 for order 3; the remainder are theoretical exclusions.

Provenance. The counts below are the outputs of the supplied archive, which contains frozen encoding code, a run plan, and producer manifests; the frozen polynomial feature kernel matches the locally inspected kernel. The main claims do not rely on these counts as evidence of strict gains.

Below, each cell a/b gives the number of observed separations in the 270 2-FWL and 110 only-3-FWL groups, respectively. For order 2, T denotes a theoretical exclusion. The 20 neither-group pairs are excluded for both orders. M1, M2 and M4 are cumulative channelwise mean moments through the indicated power.

Table 4: Monomial, order 2: archived budget and readout counts.

Dictionary	Q	Multiset	M1	M2	M4
polynomial	2	$195/\mathrm{T}$	$61/\mathrm{T}$	$190/\mathrm{T}$	$193/\mathrm{T}$
polynomial	4	$211/\mathrm{T}$	$157/\mathrm{T}$	$205/\mathrm{T}$	$205/\mathrm{T}$
polynomial	8	$233/\mathrm{T}$	$186/\mathrm{T}$	$207/\mathrm{T}$	$207/\mathrm{T}$
polynomial	16	$269/\mathrm{T}$	$210/\mathrm{T}$	$211/\mathrm{T}$	$212/\mathrm{T}$
endpoint	2	$197/\mathrm{T}$	$92/\mathrm{T}$	$192/\mathrm{T}$	$197/\mathrm{T}$
endpoint	4	$262/\mathrm{T}$	$179/\mathrm{T}$	$218/\mathrm{T}$	$232/\mathrm{T}$
endpoint	8	$270/\mathrm{T}$	$215/\mathrm{T}$	$261/\mathrm{T}$	$268/\mathrm{T}$
endpoint	16	$270/\mathrm{T}$	$266/\mathrm{T}$	$270/\mathrm{T}$	$270/\mathrm{T}$

F.1 Recorded Scale Comparison

These observations use joint $Q = 4$ and the 110 only-3-FWL pairs. The labels are those of the supplied report. Unrounded comparisons retain their numerical tolerance and do not mean exact equality. The scale artifact arrays and producer manifests must be consulted before attributing differences to a specific numerical mechanism.

Table 5: Monomial, order 3: archived budget and readout counts.

Dictionary	Q	Multiset	M1	M2	M4
polynomial	2	$207/15$	$116/14$	$193/14$	$196/14$
polynomial	4	$213/69$	$185/20$	$209/20$	$209/20$
polynomial	8	$268/109$	$192/24$	$214/24$	$214/24$
polynomial	16	$270/109$	$216/34$	$218/35$	$219/35$
endpoint	2	$207/48$	$133/15$	$198/15$	$201/17$
endpoint	4	$267/105$	$191/27$	$224/28$	$238/80$
endpoint	8	$270/109$	$233/42$	$261/76$	$269/105$
endpoint	16	$270/110$	$270/97$	$270/108$	$270/109$
joint	2	$210/88$	$134/15$	$209/15$	$209/17$
joint	4	$270/108$	$191/27$	$224/28$	$238/80$
joint	8	$270/110$	$233/42$	$261/76$	$269/105$
joint	16	$270/110$	$270/97$	$270/108$	$270/109$

Table 6: Observed separations under the five scale/comparison routes. R: rounded; U: unrounded; restored: moments returned to K units.

Basis	P	K R	K/n R	K U	K/n U	Restored U
Monomial	1	27	27	7	7	7
Monomial	2	28	28	10	10	10
Monomial	4	80	80	74	74	74

Table 7: Joint $Q = 4$ scale comparison on the 270 2-FWL pairs; route labels match the preceding table.

Basis	P	K R	K/n R	K U	K/n U	Restored U
Monomial	1	192	191	192	192	194
Monomial	2	225	224	224	224	224
Monomial	4	238	238	237	237	237

G ARCHIVED PPGN PROTOCOL AND RESULTS

Completed campaign and input provenance. The source is `reports/results 2/ppgn_dual_gpu_w16_l2`. The root manifest and all 650 job statuses report completion. The campaign contains 810 pair-model seed runs and 1,140 triple-model seed runs. The model, runner, and scheduler source hashes match those recorded in the campaign manifest. The shared `FixedInput` implementation constructs $[I, S, \dots, S^8]$, with $S = D^{-1/2}AD^{-1/2}$ and inverse degree zero at isolated vertices. Pair inputs have 9 channels. Triple inputs concatenate this dictionary at (u, v), (u, w), and (v, w), giving 27 channels. The baseline hashes the unrounded little-endian float32 rows of these same arrays and compares their multisets for one stored A/B pair. It performs no relabeling checks, separately from the neural reliability test. The archived `rrwp` label and eight-decimal metadata do not describe a row-normalized kernel or an applied rounding step: the implementation uses symmetric normalization and raw row bytes. This source correspondence establishes the input construction.

Two model architectures. The pair model adapts the PPGN regular-block structure (Maron et al., 2019). Each branch is a single pointwise affine map, implemented by a pointwise convolution (1×1 for pairs, $1\times1\times1$ for triples). The pair contraction is

$$Z(u, v) = \sum_t f_1(H(u, t)) \odot f_2(H(t, v)).$$

The third-order extension uses three branches and a common pivot:

$$Z(u, v, w) = \sum_t f_1(H(t, v, w)) \odot f_2(H(u, t, w)) \odot f_3(H(u, v, t)),$$

For both orders $k \in \{2, 3\}$, the block update is

$$H' = \mathrm{BN}\big(g\big([H, \rho_k(Z)]\big)\big), \qquad \rho_k(z) = \mathrm{sign}(z)\big((|z| + 10^{-6})^{1/k} - (10^{-6})^{1/k}\big),$$

where g is a pointwise affine projection and BN is `BatchNorm2d` or `BatchNorm3d`. The branch maps contain no ReLU. Thus the pair contraction also uses a stabilized signed root. The triple contraction loops over the shared pivot in both forward and backward passes, avoiding retention of an n^4 activation tensor. The completed models have two blocks and width 16, with recorded parameter counts 3,200 (pair) and 6,320 (triple). They are not parameter-matched.

Implemented readout. After each block, pair pooling assigns node u the concatenation $[H(u, u), \sum_{v\neq u} H(u, v)/\max(n-1, 1)]$. Triple pooling assigns it five channel vectors:

$$\Big[H(u,u,u),\ \frac{1}{n}\sum_t H(u,t,t),\ \frac{1}{n}\sum_t H(t,u,t),\ \frac{1}{n}\sum_t H(t,t,u),$$
$$\frac{1}{3n^2}\sum_{v,w}\big(H(u,v,w) + H(v,u,w) + H(v,w,u)\big)\Big].$$

These triple summaries overlap: the diagonal planes include the main diagonal, and the marginal includes repeated coordinates. They are not disjoint equality-pattern averages. A separate affine map sends each block's node summary to 16 channels. The model sums these contributions over blocks, takes their node mean, and applies a final affine map to 16 outputs. There is no ReLU in this head. The third-order model is a shared-pivot extension of this backbone, not a complete implementation of iterative 3-FWL.

Training. A separate model is trained for each graph pair and seed 100, 200, or 300. Adam (Kingma & Ba, 2015) uses learning rate 10^{-4} and weight decay 10^{-5}, with cosine embedding loss (target -1, margin zero), gradient norm clipping at 1, and ReduceLROnPlateau. Training lasts at most 20 epochs and stops when the epoch-average loss falls below 0.2. Gradient accumulation gives an effective batch of 32 graphs; the requested physical batch is 32. All 810 pair runs use 32. Triple runs use 32 in 1,065 cases, 16 in 54, 8 in 18, and 4 in three cases, with effective batch 32. Changing the physical batch also changes the training-time BatchNorm statistics; gradient accumulation does not remove this difference. All 1,950 completed records report float32. Reliability graphs are never used in training. Only the trained model is evaluated: there are no fresh-model, normalization-only, independent-pivot, basis, or depth experiments in this protocol.

Neural decisions. For 32 paired relabelings, let $d_i = z(G_i) - z(H_i)$ and let S be their sample covariance. We follow the BREC code convention (Wang & Zhang, 2024)

$$T = \bar{d}^{\mathsf{T}} S^{+} \bar{d}$$

without an additional sample-count factor. The main score must exceed 72.34, while the score on the supplied isomorphic reliability pairs must be below 72.34; approximately equal main and reliability scores are not accepted as a distinction. Covariance-pseudoinverse and centered-data SVD implementations use matched spectral cutoffs and must agree within relative tolerance 10^{-3} and absolute tolerance 10^{-5}. Float32 covariance eigenvalues below $d\epsilon\lambda_{\max}$ are truncated, where $d = 16$. A zero retained rank with a mean difference norm above 10^{-7} is unresolved, not a confirmed nonseparation. Scorer disagreement, nonfinite values, reliability failures, and reference-upper-bound violations are recorded explicitly and excluded from confirmed decisions. Scorer agreement alone is not evidence of valid graph discrimination. Both main and reliability outputs are saved for independent inspection and rescoring.

Coverage and reporting. The executed population is 270 2-FWL pairs for the pair model and 380 2-FWL or only-3-FWL pairs for the triple model, for each seed. The remaining 130 and 20 pairs, respectively, are excluded rather than measured nonseparations. Adaptive physical batches are recorded explicitly. No unresolved comparisons occur in the supplied summaries. Input and trained decisions are joined by pair ID within each reference group. Additional and not-retained distinctions are computed only on pairs with valid decisions for both methods. Per-seed counts and pair IDs are reported; seeds are not combined into a union score. Inference is limited to these per-pair fitting experiments.

G.1 Recorded separation counts

Table 8: Archived PPGN campaign. Counts are per seed on common valid pairs, using unrounded multisets of the same symmetric-power inputs as the models. Excluded reference groups were not run and are not counted as empirical failures.

Model	Group	Seed	Input	Trained	Added	Lost	Unresolved
Pair	2-FWL	100	254	256	11	9	0
Pair	2-FWL	200	254	260	13	7	0
Pair	2-FWL	300	254	256	11	9	0
Third-order	2-FWL	100	270	270	0	0	0
Third-order	Only-3-FWL	100	109	97	0	12	0
Third-order	2-FWL	200	270	269	0	1	0
Third-order	Only-3-FWL	200	109	92	0	17	0
Third-order	2-FWL	300	270	269	0	1	0
Third-order	Only-3-FWL	300	109	85	0	24	0

G.2 Recorded additional distinctions

Pair IDs are zero-based. The triple model has no recorded gains at any seed.

Pair model, seed 100. 268, 271, 281, 284, 286, 294, 297, 307, 308, 310, 314.

Pair model, seed 200. 260, 262, 268, 271, 281, 284, 286, 294, 297, 307, 308, 310, 314.

Pair model, seed 300. 262, 268, 271, 281, 284, 294, 297, 307, 308, 310, 314.

H ZINC Subset Object Discrimination

The source is `reports/zinc_subset_summary.json`, identifying the PyG ZINC subset (Irwin et al., 2012; Dwivedi et al., 2023; Fey & Lenssen, 2019), all 12,000 graphs, and no node-count or distance cutoff. The float32 v13 run evaluates five channel-prefix widths $d = 6, 8, 10, 12, 14$ and five routes on nodes, existing unordered edges, and all unordered distinct pairs: 75 configurations in total.

These are width settings, not complete polynomial-degree budgets. Mixed and each Wider route use $2d$ channels per orientation; Powers and Endpoints use d.

Every configuration reports 12,000 reliable graphs and 12,000 valid RC measurements, with no inexact MIS solves. The protocol uses three relabeling checks, absolute tolerance 10^{-8} and relative tolerance 10^{-6}, without column standardization. Means below are graph macro averages. Unresolved counts graphs with a missed cross-orbit comparison; a high average RC does not imply full recovery on each graph.

Table 9: ZINC subset nodes: complete width scan. PR and RC are macro means; unresolved is a graph count out of 12,000.

d	Route	PR	RC	Unresolved
6	Powers	0.997112	0.972637	4400
6	Endpoints	0.999063	0.990688	1531
6	Mixed	0.999374	0.993535	1222
6	Wider Powers	0.999982	0.999812	40
6	Wider Endpoints	0.999932	0.999343	106
8	Powers	0.999494	0.994769	976
8	Endpoints	0.999063	0.990688	1531
8	Mixed	0.999519	0.995049	930
8	Wider Powers	0.999997	0.999976	2
8	Wider Endpoints	0.999982	0.999832	26
10	Powers	0.999899	0.998885	227
10	Endpoints	0.999759	0.997578	399
10	Mixed	0.999906	0.998939	219
10	Wider Powers	0.999997	0.999976	2
10	Wider Endpoints	0.999995	0.999968	6
12	Powers	0.999982	0.999812	40
12	Endpoints	0.999932	0.999343	106
12	Mixed	0.999986	0.999848	36
12	Wider Powers	0.999997	0.999976	2
12	Wider Endpoints	0.999995	0.999968	6
14	Powers	0.999997	0.999967	5
14	Endpoints	0.999932	0.999343	106
14	Mixed	1.000000	0.999991	3
14	Wider Powers	0.999997	0.999976	2
14	Wider Endpoints	1.000000	1.000000	0

Table 10: ZINC subset edges: complete width scan. PR and RC are macro means; unresolved is a graph count out of 12,000.

d	Route	PR	RC	Unresolved
6	Powers	0.995486	0.954301	6327
6	Endpoints	0.999577	0.995553	713
6	Mixed	0.999785	0.997590	453
6	Wider Powers	0.999968	0.999679	64
6	Wider Endpoints	0.999968	0.999652	55
8	Powers	0.999213	0.991879	1458
8	Endpoints	0.999577	0.995553	713
8	Mixed	0.999820	0.998017	384
8	Wider Powers	0.999989	0.999931	9
8	Wider Endpoints	0.999989	0.999912	15
10	Powers	0.999840	0.998292	327
10	Endpoints	0.999884	0.998770	186
10	Mixed	0.999970	0.999657	75
10	Wider Powers	0.999989	0.999931	9
10	Wider Endpoints	0.999997	0.999983	3
12	Powers	0.999968	0.999679	64
12	Endpoints	0.999968	0.999652	55
12	Mixed	0.999998	0.999961	10
12	Wider Powers	0.999989	0.999931	9
12	Wider Endpoints	0.999997	0.999983	3
14	Powers	0.999988	0.999900	17
14	Endpoints	0.999968	0.999652	55
14	Mixed	0.999999	0.999970	8
14	Wider Powers	0.999989	0.999931	9
14	Wider Endpoints	1.000000	1.000000	0

Table 11: ZINC subset pairs: complete width scan. PR and RC are macro means; unresolved is a graph count out of 12,000.

d	Route	PR	RC	Unresolved
6	Powers	0.897960	0.623108	11969
6	Endpoints	0.999468	0.973019	8435
6	Mixed	0.999956	0.995531	1453
6	Wider Powers	0.998263	0.957866	11005
6	Wider Endpoints	0.999658	0.988718	7965
8	Powers	0.967451	0.802499	11774
8	Endpoints	0.999468	0.973019	8435
8	Mixed	0.999980	0.997814	1148
8	Wider Powers	0.999724	0.979576	10629
8	Wider Endpoints	0.999670	0.989598	7942
10	Powers	0.991732	0.907595	11406
10	Endpoints	0.999619	0.985532	8056
10	Mixed	0.999998	0.999726	225
10	Wider Powers	0.999740	0.980301	10597
10	Wider Endpoints	0.999673	0.989836	7941
12	Powers	0.998263	0.957866	11005
12	Endpoints	0.999658	0.988718	7965
12	Mixed	1.000000	0.999968	38
12	Wider Powers	0.999740	0.980306	10596
12	Wider Endpoints	0.999673	0.989836	7941
14	Powers	0.999549	0.975404	10735
14	Endpoints	0.999658	0.988718	7965
14	Mixed	1.000000	0.999987	23
14	Wider Powers	0.999740	0.980306	10596
14	Wider Endpoints	0.999675	0.989896	7941

H.1 Matched Comparisons

All paired comparisons below use the same 12,000 graphs. Differences are first route minus second route, with graph-bootstrap 95% intervals from the supplied paired summaries. These intervals describe variation across graphs, not training seeds. No task-prediction result is implied.

Table 12: RC differences at $d = 14$; Mixed and Wider routes have 28 channels.

Domain	Contrast	Mean Δ	CI low	CI high
nodes	Endpoints – Powers	-0.000624	-0.000790	-0.000472
nodes	Mixed – Wider Powers	+0.000015	-0.000014	+0.000058
nodes	Wider Endpoints – Wider Powers	+0.000024	+0.000000	+0.000064
edges	Endpoints – Powers	-0.000248	-0.000377	-0.000123
edges	Mixed – Wider Powers	+0.000039	-0.000016	+0.000116
edges	Wider Endpoints – Wider Powers	+0.000069	+0.000020	+0.000141
pairs	Endpoints – Powers	+0.013314	+0.012535	+0.014070
pairs	Wider Endpoints – Wider Powers	+0.009590	+0.008897	+0.010262

At $d = 14$, 10,735 graphs retain unresolved pair comparisons under Powers, 10,596 under Wider Powers, 7,965 under Endpoints, 7,941 under Wider Endpoints, and 23 under Mixed. On nodes and edges, the Mixed versus Wider Powers RC intervals include zero. Wider Endpoints reports no unresolved graphs for nodes or edges at this width; Mixed is not uniformly the strongest route across object domains.

Cost scope. The summary records 693.13 seconds for the reported invocation and 11,399.09 summed graph-seconds, including 14.20 summed orbit-seconds. Because the run configuration has resume enabled, the invocation timer is not asserted to be the end-to-end campaign duration. The reported maximum worker high-water RSS is 0.06524 GiB; it is not aggregate concurrent RAM. The run requests 100 single-thread workers and a 100-GiB admission budget. These are shared-workload measurements, not isolated costs for each encoding route.

I QM9 Object Discrimination

The source is `reports/qm9_summary.json`. The completed float32 v13 run covers all 130,831 PyG QM9 graphs (Ramakrishnan et al., 2014; Fey & Lenssen, 2019), without a node-count cutoff or distance gate. Objects are nodes, existing unordered edges, and all unordered distinct vertex pairs. The latter replaces the historical `other_pairs` subset; values across these two domains are not interpreted as changes under an identical protocol. Five channel-prefix widths $d = 6, 8, 10, 12, 14$ and five routes give 75 configurations. Powers and Endpoints use d channels per orientation; Mixed and Wider routes use $2d$. These widths are not complete degree budgets.

The protocol uses three permutation checks, absolute tolerance 10^{-8}, relative tolerance 10^{-6}, and no column standardization. All MIS solves are exact. Wider Powers excludes two graphs at $d = 10$ and three at $d = 12, 14$ in each of the edge and pair domains because relabeled comparison masks change. All other configurations retain 130,831 reliable graphs. Tables report route-specific graph macro means and their eligible counts. For nodes, PR excludes one graph without cross-orbit comparisons; for edges it excludes seven such graphs in addition to reliability exclusions. RC remains defined for those degenerate cases. Unresolved counts eligible graphs with missed cross-orbit comparisons. Rounded PR or RC of one does not by itself imply zero unresolved graphs.

Table 13: QM9 nodes: complete width scan. N_C and N_R are valid RC and PR graph counts.

d	Route	PR	RC	N_C	N_R	Unresolved
6	Powers	0.998337	0.994060	130831	130830	10108
6	Endpoints	0.999784	0.999192	130831	130830	1317
6	Mixed	0.999922	0.999714	130831	130830	565
6	Wider Powers	0.999974	0.999820	130831	130830	138
6	Wider Endpoints	0.999972	0.999933	130831	130830	32
8	Powers	0.999936	0.999686	130831	130830	407
8	Endpoints	0.999784	0.999192	130831	130830	1317
8	Mixed	0.999972	0.999896	130831	130830	204
8	Wider Powers	0.999974	0.999820	130831	130830	138
8	Wider Endpoints	0.999974	0.999940	130831	130830	21
10	Powers	0.999974	0.999820	130831	130830	139
10	Endpoints	0.999964	0.999895	130831	130830	84
10	Mixed	0.999999	0.999996	130831	130830	4
10	Wider Powers	0.999974	0.999820	130831	130830	138
10	Wider Endpoints	0.999974	0.999941	130831	130830	19
12	Powers	0.999974	0.999820	130831	130830	138
12	Endpoints	0.999972	0.999933	130831	130830	32
12	Mixed	0.999999	0.999996	130831	130830	4
12	Wider Powers	0.999974	0.999820	130831	130830	138
12	Wider Endpoints	0.999974	0.999941	130831	130830	19
14	Powers	0.999974	0.999820	130831	130830	138
14	Endpoints	0.999972	0.999933	130831	130830	32
14	Mixed	0.999999	0.999996	130831	130830	4
14	Wider Powers	0.999974	0.999820	130831	130830	138
14	Wider Endpoints	0.999974	0.999941	130831	130830	19

Table 14: QM9 edges: complete width scan. N_C and N_R are valid RC and PR graph counts.

d	Route	PR	RC	N_C	N_R	Unresolved
6	Powers	0.998191	0.992342	130831	130824	11492
6	Endpoints	0.999950	0.999801	130831	130824	150
6	Mixed	0.999999	0.999995	130831	130824	11
6	Wider Powers	0.999926	0.999555	130831	130824	497
6	Wider Endpoints	0.999961	0.999870	130831	130824	85
8	Powers	0.999888	0.999417	130831	130824	770
8	Endpoints	0.999950	0.999801	130831	130824	150
8	Mixed	1.000000	0.999998	130831	130824	4
8	Wider Powers	0.999926	0.999555	130831	130824	497
8	Wider Endpoints	0.999961	0.999871	130831	130824	83
10	Powers	0.999926	0.999554	130831	130824	498
10	Endpoints	0.999959	0.999857	130831	130824	96
10	Mixed	1.000000	0.999998	130831	130824	4
10	Wider Powers	0.999927	0.999556	130829	130822	495
10	Wider Endpoints	0.999961	0.999871	130831	130824	83
12	Powers	0.999926	0.999555	130831	130824	497
12	Endpoints	0.999961	0.999870	130831	130824	85
12	Mixed	1.000000	0.999998	130831	130824	4
12	Wider Powers	0.999927	0.999557	130828	130821	494
12	Wider Endpoints	0.999961	0.999871	130831	130824	83
14	Powers	0.999926	0.999555	130831	130824	497
14	Endpoints	0.999961	0.999870	130831	130824	85
14	Mixed	1.000000	0.999998	130831	130824	4
14	Wider Powers	0.999927	0.999557	130828	130821	494
14	Wider Endpoints	0.999961	0.999871	130831	130824	83

Table 15: QM9 pairs: complete width scan. N_C and N_R are valid RC and PR graph counts.

d	Route	PR	RC	N_C	N_R	Unresolved
6	Powers	0.994401	0.873434	130831	130831	129280
6	Endpoints	0.998034	0.974562	130831	130831	39668
6	Mixed	0.999998	0.999938	130831	130831	571
6	Wider Powers	0.997422	0.910424	130831	130831	128740
6	Wider Endpoints	0.998087	0.975929	130831	130831	38740
8	Powers	0.997334	0.908994	130831	130831	128762
8	Endpoints	0.998034	0.974562	130831	130831	39668
8	Mixed	0.999999	0.999969	130831	130831	257
8	Wider Powers	0.997422	0.910424	130831	130831	128740
8	Wider Endpoints	0.998088	0.975943	130831	130831	38733
10	Powers	0.997422	0.910417	130831	130831	128740
10	Endpoints	0.998084	0.975860	130831	130831	38764
10	Mixed	0.999999	0.999985	130831	130831	38
10	Wider Powers	0.997422	0.910424	130829	130829	128738
10	Wider Endpoints	0.998088	0.975945	130831	130831	38731
12	Powers	0.997422	0.910424	130831	130831	128740
12	Endpoints	0.998087	0.975929	130831	130831	38740
12	Mixed	0.999999	0.999986	130831	130831	27
12	Wider Powers	0.997422	0.910424	130828	130828	128737
12	Wider Endpoints	0.998088	0.975945	130831	130831	38731
14	Powers	0.997422	0.910424	130831	130831	128740
14	Endpoints	0.998087	0.975929	130831	130831	38740
14	Mixed	0.999999	0.999986	130831	130831	27
14	Wider Powers	0.997422	0.910424	130828	130828	128737
14	Wider Endpoints	0.998088	0.975945	130831	130831	38731

I.1 Matched Comparisons

Differences below use the common eligible graphs for each contrast, rather than subtracting route means with different populations. Intervals are supplied 95% graph-bootstrap intervals, not variation across training seeds. At $d = 14$, the pair domain retains 128,740 unresolved graphs for Powers, 38,740 for Endpoints, and 27 for Mixed. Wider Powers has 128,737 unresolved graphs among 130,828 eligible graphs. On nodes and edges, the paired Mixed–Wider Powers RC differences are 0.000176 and 0.000440, respectively. These object-discrimination measurements do not establish a downstream prediction improvement.

Table 16: QM9 paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
nodes	Endpoints – Powers	130831	0.000113	0.000069	0.000155
nodes	Mixed – Wider Powers	130831	0.000176	0.000146	0.000207
nodes	Wider Endpoints – Wider Powers	130831	0.000121	0.000079	0.000163
edges	Endpoints – Powers	130831	0.000315	0.000258	0.000371
edges	Mixed – Wider Powers	130828	0.000440	0.000397	0.000484
edges	Wider Endpoints – Wider Powers	130828	0.000314	0.000258	0.000371
pairs	Endpoints – Powers	130831	0.065505	0.065048	0.065957
pairs	Wider Endpoints – Wider Powers	130828	0.065520	0.065064	0.065970

Cost scope. The non-resumed invocation reports 11,248.40 seconds elapsed and 90,370.26 summed graph-seconds, including 135.89 orbit-seconds. It requests 12 single-thread workers and a 100-GiB memory admission budget. Maximum worker high-water RSS is 0.20541 GiB, not total concurrent RAM. These costs cover the shared workload across routes and widths.

J Additional Object Discrimination Results

Six reports in `reports/object_summaries` provide 400 configurations: ENZYMES, MUTAG, NCI1, and NCI109 each cover nodes, edges, and all unordered distinct pairs; PROTEINS and molhiv cover nodes and edges only. The first five are TU datasets (Morris et al., 2020); molhiv is the OGB molecular benchmark (Hu et al., 2020; Wu et al., 2018). All use float32, channel-prefix widths $d = 6, 8, 10, 12, 14$, five routes, three permutation checks, tolerances 10^{-8} (absolute) and 10^{-6} (relative), and no distance gate. Mixed and Wider routes use $2d$ channels per orientation; Powers and Endpoints use d.

PROTEINS includes all 1,113 graphs without a node-count cutoff. It has 50 node/edge configurations and no pair-domain measurements. All six reports contain only exact MIS solves. Numerical reliability exclusions remain route-specific. All six runs report completion.

Tables give graph macro PR (pairwise resolution) and RC (representative capacity), their respective valid populations N_R and N_C, unreliable count U, inexact MIS count I, and the reported unresolved count M. The latter counts missed cross-orbit comparisons among reliable graphs and need not use the exact-MIS RC population. Paired RC comparisons use common eligible graphs and supplied 95% graph-bootstrap confidence intervals. No missing value is replaced by zero.

J.1 ENZYMES

The report processes 600 graphs and 75 configurations.

Table 17: ENZYMES, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.999195	0.988532	597	600	0	0	62
6	Endpoints	0.999635	0.992863	597	600	0	0	46
6	Mixed	0.999661	0.993266	597	600	0	0	42
6	Wider Powers	0.999872	0.997076	597	600	0	0	16
6	Wider Endpoints	0.999917	0.997404	597	600	0	0	16
8	Powers	0.999631	0.993633	597	600	0	0	33
8	Endpoints	0.999635	0.992863	597	600	0	0	46
8	Mixed	0.999698	0.994115	597	600	0	0	32
8	Wider Powers	0.999921	0.998404	597	600	0	0	9
8	Wider Endpoints	0.999954	0.998287	597	600	0	0	11
10	Powers	0.999808	0.996002	595	598	2	0	21
10	Endpoints	0.999830	0.995844	597	600	0	0	26
10	Mixed	0.999863	0.996471	595	598	2	0	20
10	Wider Powers	0.999933	0.998865	597	600	0	0	7
10	Wider Endpoints	0.999972	0.998852	597	600	0	0	6
12	Powers	0.999872	0.997076	597	600	0	0	16
12	Endpoints	0.999917	0.997404	597	600	0	0	16
12	Mixed	0.999922	0.997525	597	600	0	0	15
12	Wider Powers	0.999946	0.999231	595	598	2	0	4
12	Wider Endpoints	0.999972	0.998852	597	600	0	0	6
14	Powers	0.999904	0.997855	597	600	0	0	12
14	Endpoints	0.999917	0.997404	597	600	0	0	16
14	Mixed	0.999952	0.998271	597	600	0	0	11
14	Wider Powers	0.999946	0.999290	596	599	1	0	5
14	Wider Endpoints	0.999981	0.999239	597	600	0	0	5

Table 18: ENZYMES, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.999179	0.984864	597	600	0	0	98
6	Endpoints	0.999584	0.991312	597	600	0	0	55
6	Mixed	0.999794	0.994075	597	600	0	0	42
6	Wider Powers	0.999911	0.996765	597	600	0	0	18
6	Wider Endpoints	0.999765	0.995893	597	600	0	0	30
8	Powers	0.999634	0.991819	597	600	0	0	53
8	Endpoints	0.999584	0.991312	597	600	0	0	55
8	Mixed	0.999799	0.994281	597	600	0	0	40
8	Wider Powers	0.999967	0.998566	596	599	1	0	12
8	Wider Endpoints	0.999795	0.996725	597	600	0	0	24
10	Powers	0.999828	0.995179	597	600	0	0	26
10	Endpoints	0.999705	0.994199	597	600	0	0	36
10	Mixed	0.999898	0.996427	597	600	0	0	22
10	Wider Powers	0.999982	0.999145	597	600	0	0	8
10	Wider Endpoints	0.999808	0.997335	597	600	0	0	21
12	Powers	0.999911	0.996765	597	600	0	0	18
12	Endpoints	0.999765	0.995893	597	600	0	0	30
12	Mixed	0.999947	0.997817	597	600	0	0	15
12	Wider Powers	0.999988	0.999395	597	600	0	0	7
12	Wider Endpoints	0.999808	0.997335	597	600	0	0	21
14	Powers	0.999950	0.997946	597	600	0	0	14
14	Endpoints	0.999765	0.995893	597	600	0	0	30
14	Mixed	0.999950	0.997969	597	600	0	0	14
14	Wider Powers	0.999992	0.999678	596	599	1	0	5
14	Wider Endpoints	0.999816	0.997684	597	600	0	0	20

Table 19: ENZYMES, pairs: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.911599	0.760155	597	600	0	0	511
6	Endpoints	0.998743	0.980220	597	600	0	0	186
6	Mixed	0.999977	0.991282	597	600	0	0	47
6	Wider Powers	0.983065	0.951575	594	597	3	0	262
6	Wider Endpoints	0.998774	0.988143	597	600	0	0	166
8	Powers	0.957019	0.861056	597	600	0	0	417
8	Endpoints	0.998743	0.980220	597	600	0	0	186
8	Mixed	0.999982	0.992962	597	600	0	0	46
8	Wider Powers	0.987314	0.973077	591	594	6	0	144
8	Wider Endpoints	0.998777	0.989690	597	600	0	0	162
10	Powers	0.975728	0.919898	597	600	0	0	344
10	Endpoints	0.998765	0.985411	597	600	0	0	173
10	Mixed	0.999994	0.996120	597	600	0	0	26
10	Wider Powers	0.988135	0.978200	593	596	4	0	105
10	Wider Endpoints	0.998778	0.990697	597	600	0	0	159
12	Powers	0.983065	0.951575	594	597	3	0	262
12	Endpoints	0.998774	0.988143	597	600	0	0	166
12	Mixed	0.999997	0.997577	597	600	0	0	16
12	Wider Powers	0.988207	0.980704	589	592	8	0	87
12	Wider Endpoints	0.998778	0.990697	597	600	0	0	159
14	Powers	0.986075	0.966190	595	598	2	0	198
14	Endpoints	0.998774	0.988143	597	600	0	0	166
14	Mixed	0.999998	0.997936	597	600	0	0	15
14	Wider Powers	0.988308	0.981466	593	596	4	0	79
14	Wider Endpoints	0.998779	0.991364	597	600	0	0	158

Table 20: ENZYMES: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	600	-0.002053	-0.003238	-0.001019
edges	Mixed – Powers	600	0.000023	0.000000	0.000068
edges	Mixed – Wider Powers	599	-0.001321	-0.002440	-0.000461
edges	Wider Endpoints – Wider Powers	599	-0.001808	-0.002993	-0.000840
nodes	Endpoints – Powers	600	-0.000450	-0.001266	0.000635
nodes	Mixed – Powers	600	0.000417	0.000000	0.001250
nodes	Mixed – Wider Powers	599	-0.000841	-0.002152	0.000505
nodes	Wider Endpoints – Wider Powers	599	-0.000052	-0.000838	0.001018
pairs	Endpoints – Powers	598	0.022376	0.014665	0.030647
pairs	Mixed – Powers	598	0.031821	0.024291	0.039917
pairs	Mixed – Wider Powers	596	0.018009	0.011138	0.025667
pairs	Wider Endpoints – Wider Powers	596	0.010939	0.003910	0.018503

J.2 MUTAG

The report processes 188 graphs and 75 configurations.

Table 21: MUTAG, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.990764	0.938775	188	188	0	0	92
6	Endpoints	0.996853	0.975359	188	188	0	0	42
6	Mixed	0.997140	0.977362	188	188	0	0	39
6	Wider Powers	0.999582	0.996575	188	188	0	0	9
6	Wider Endpoints	0.999561	0.996333	188	188	0	0	10
8	Powers	0.997182	0.977846	188	188	0	0	39
8	Endpoints	0.996853	0.975359	188	188	0	0	42
8	Mixed	0.997182	0.977846	188	188	0	0	39
8	Wider Powers	0.999944	0.999325	188	188	0	0	2
8	Wider Endpoints	0.999872	0.998550	188	188	0	0	5
10	Powers	0.998822	0.990443	188	188	0	0	22
10	Endpoints	0.998677	0.989199	188	188	0	0	24
10	Mixed	0.998822	0.990443	188	188	0	0	22
10	Wider Powers	0.999944	0.999325	188	188	0	0	2
10	Wider Endpoints	0.999944	0.999325	188	188	0	0	2
12	Powers	0.999582	0.996575	188	188	0	0	9
12	Endpoints	0.999561	0.996333	188	188	0	0	10
12	Mixed	0.999582	0.996575	188	188	0	0	9
12	Wider Powers	0.999944	0.999325	188	188	0	0	2
12	Wider Endpoints	0.999944	0.999325	188	188	0	0	2
14	Powers	0.999872	0.998550	188	188	0	0	5
14	Endpoints	0.999561	0.996333	188	188	0	0	10
14	Mixed	0.999872	0.998550	188	188	0	0	5
14	Wider Powers	0.999944	0.999325	188	188	0	0	2
14	Wider Endpoints	0.999944	0.999325	188	188	0	0	2

Table 22: MUTAG, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.987628	0.908315	188	188	0	0	114
6	Endpoints	0.998464	0.985404	188	188	0	0	27
6	Mixed	0.998691	0.987177	188	188	0	0	26
6	Wider Powers	0.999632	0.996558	188	188	0	0	10
6	Wider Endpoints	0.999907	0.998768	188	188	0	0	5
8	Powers	0.997292	0.975305	188	188	0	0	42
8	Endpoints	0.998464	0.985404	188	188	0	0	27
8	Mixed	0.998770	0.987842	188	188	0	0	24
8	Wider Powers	1.000000	1.000000	188	188	0	0	0
8	Wider Endpoints	0.999985	0.999795	188	188	0	0	1
10	Powers	0.998880	0.989210	188	188	0	0	23
10	Endpoints	0.999489	0.995064	188	188	0	0	14
10	Mixed	0.999568	0.995729	188	188	0	0	12
10	Wider Powers	1.000000	1.000000	188	188	0	0	0
10	Wider Endpoints	1.000000	1.000000	188	188	0	0	0
12	Powers	0.999632	0.996558	188	188	0	0	10
12	Endpoints	0.999907	0.998768	188	188	0	0	5
12	Mixed	0.999907	0.998768	188	188	0	0	5
12	Wider Powers	1.000000	1.000000	188	188	0	0	0
12	Wider Endpoints	1.000000	1.000000	188	188	0	0	0
14	Powers	0.999927	0.999106	188	188	0	0	3
14	Endpoints	0.999907	0.998768	188	188	0	0	5
14	Mixed	0.999927	0.999106	188	188	0	0	3
14	Wider Powers	1.000000	1.000000	188	188	0	0	0
14	Wider Endpoints	1.000000	1.000000	188	188	0	0	0

Table 23: MUTAG, pairs: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.984604	0.837586	188	188	0	0	173
6	Endpoints	0.996116	0.891738	188	188	0	0	89
6	Mixed	0.999807	0.989286	188	188	0	0	41
6	Wider Powers	0.999332	0.978889	188	188	0	0	104
6	Wider Endpoints	0.996766	0.929131	188	188	0	0	58
8	Powers	0.997726	0.953631	188	188	0	0	141
8	Endpoints	0.996116	0.891738	188	188	0	0	89
8	Mixed	0.999938	0.994875	188	188	0	0	31
8	Wider Powers	0.999370	0.980461	188	188	0	0	103
8	Wider Endpoints	0.996823	0.933303	188	188	0	0	53
10	Powers	0.999128	0.974028	188	188	0	0	113
10	Endpoints	0.996567	0.916103	188	188	0	0	71
10	Mixed	0.999986	0.998454	188	188	0	0	14
10	Wider Powers	0.999370	0.980461	188	188	0	0	103
10	Wider Endpoints	0.996838	0.934766	188	188	0	0	50
12	Powers	0.999332	0.978889	188	188	0	0	104
12	Endpoints	0.996766	0.929131	188	188	0	0	58
12	Mixed	0.999994	0.999400	188	188	0	0	8
12	Wider Powers	0.999370	0.980461	188	188	0	0	103
12	Wider Endpoints	0.996838	0.934766	188	188	0	0	50
14	Powers	0.999368	0.980234	188	188	0	0	104
14	Endpoints	0.996766	0.929131	188	188	0	0	58
14	Mixed	0.999996	0.999572	188	188	0	0	5
14	Wider Powers	0.999370	0.980461	188	188	0	0	103
14	Wider Endpoints	0.996838	0.934766	188	188	0	0	50

Table 24: MUTAG: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	188	-0.000338	-0.000847	0.000000
edges	Mixed – Powers	188	0.000000	0.000000	0.000000
edges	Mixed – Wider Powers	188	-0.000894	-0.002085	0.000000
edges	Wider Endpoints – Wider Powers	188	0.000000	0.000000	0.000000
nodes	Endpoints – Powers	188	-0.002217	-0.003758	-0.000904
nodes	Mixed – Powers	188	0.000000	0.000000	0.000000
nodes	Mixed – Wider Powers	188	-0.000775	-0.001792	0.000000
nodes	Wider Endpoints – Wider Powers	188	0.000000	0.000000	0.000000
pairs	Endpoints – Powers	188	-0.051103	-0.074327	-0.029746
pairs	Mixed – Powers	188	0.019338	0.015137	0.023743
pairs	Mixed – Wider Powers	188	0.019111	0.014897	0.023651
pairs	Wider Endpoints – Wider Powers	188	-0.045695	-0.069222	-0.023895

J.3 NCI1

The report processes 4,110 graphs and 75 configurations.

Table 25: NCI1, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.993539	0.942709	4108	4110	0	0	2227
6	Endpoints	0.997414	0.976591	4108	4110	0	0	1068
6	Mixed	0.997741	0.978394	4108	4110	0	0	1022
6	Wider Powers	0.999538	0.994990	4108	4110	0	0	224
6	Wider Endpoints	0.999408	0.994473	4108	4110	0	0	240
8	Powers	0.997667	0.978826	4108	4110	0	0	982
8	Endpoints	0.997414	0.976591	4108	4110	0	0	1068
8	Mixed	0.997860	0.979625	4108	4110	0	0	957
8	Wider Powers	0.999854	0.997947	4106	4108	2	0	67
8	Wider Endpoints	0.999657	0.996788	4108	4110	0	0	112
10	Powers	0.999013	0.990487	4108	4110	0	0	411
10	Endpoints	0.998857	0.989536	4108	4110	0	0	443
10	Mixed	0.999096	0.990853	4108	4110	0	0	404
10	Wider Powers	0.999932	0.998721	4106	4108	2	0	45
10	Wider Endpoints	0.999766	0.997874	4108	4110	0	0	69
12	Powers	0.999538	0.994990	4108	4110	0	0	224
12	Endpoints	0.999408	0.994473	4108	4110	0	0	240
12	Mixed	0.999578	0.995184	4108	4110	0	0	220
12	Wider Powers	0.999960	0.999215	4102	4104	6	0	30
12	Wider Endpoints	0.999766	0.997874	4108	4110	0	0	69
14	Powers	0.999758	0.997110	4107	4109	1	0	108
14	Endpoints	0.999408	0.994473	4108	4110	0	0	240
14	Mixed	0.999781	0.997242	4107	4109	1	0	103
14	Wider Powers	0.999972	0.999476	4099	4101	9	0	23
14	Wider Endpoints	0.999820	0.998477	4108	4110	0	0	48

Table 26: NCI1, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.992224	0.927429	4105	4110	0	0	2496
6	Endpoints	0.998485	0.986778	4105	4110	0	0	567
6	Mixed	0.998736	0.987973	4105	4110	0	0	544
6	Wider Powers	0.999422	0.994385	4105	4110	0	0	247
6	Wider Endpoints	0.999588	0.996264	4105	4110	0	0	127
8	Powers	0.997146	0.974103	4105	4110	0	0	1194
8	Endpoints	0.998485	0.986778	4105	4110	0	0	567
8	Mixed	0.998792	0.988573	4105	4110	0	0	526
8	Wider Powers	0.999818	0.998105	4098	4103	7	0	76
8	Wider Endpoints	0.999723	0.997564	4105	4110	0	0	76
10	Powers	0.998804	0.988933	4105	4110	0	0	514
10	Endpoints	0.999262	0.993529	4105	4110	0	0	268
10	Mixed	0.999452	0.994478	4105	4110	0	0	241
10	Wider Powers	0.999912	0.998771	4098	4103	7	0	44
10	Wider Endpoints	0.999794	0.998214	4105	4110	0	0	53
12	Powers	0.999422	0.994385	4105	4110	0	0	247
12	Endpoints	0.999588	0.996264	4105	4110	0	0	127
12	Mixed	0.999728	0.996826	4105	4110	0	0	115
12	Wider Powers	0.999926	0.999014	4100	4105	5	0	40
12	Wider Endpoints	0.999794	0.998214	4105	4110	0	0	53
14	Powers	0.999692	0.996697	4105	4110	0	0	125
14	Endpoints	0.999588	0.996264	4105	4110	0	0	127
14	Mixed	0.999730	0.996873	4105	4110	0	0	114
14	Wider Powers	0.999938	0.999178	4104	4109	1	0	37
14	Wider Endpoints	0.999839	0.998649	4105	4110	0	0	41

Table 27: NCI1, pairs: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.850607	0.581919	4110	4110	0	0	4053
6	Endpoints	0.996727	0.896801	4110	4110	0	0	2974
6	Mixed	0.999639	0.973792	4110	4110	0	0	1296
6	Wider Powers	0.971062	0.891751	4110	4110	0	0	3719
6	Wider Endpoints	0.997161	0.927471	4110	4110	0	0	2647
8	Powers	0.926217	0.749524	4110	4110	0	0	3973
8	Endpoints	0.996727	0.896801	4110	4110	0	0	2974
8	Mixed	0.999807	0.983489	4110	4110	0	0	1122
8	Wider Powers	0.980819	0.928767	4091	4091	19	0	3616
8	Wider Endpoints	0.997213	0.931436	4110	4110	0	0	2594
10	Powers	0.957635	0.844226	4110	4110	0	0	3812
10	Endpoints	0.997046	0.918966	4110	4110	0	0	2734
10	Mixed	0.999933	0.993665	4110	4110	0	0	474
10	Wider Powers	0.983548	0.939331	4087	4087	23	0	3587
10	Wider Endpoints	0.997234	0.933264	4110	4110	0	0	2577
12	Powers	0.971062	0.891751	4110	4110	0	0	3719
12	Endpoints	0.997161	0.927471	4110	4110	0	0	2647
12	Mixed	0.999977	0.997112	4110	4110	0	0	255
12	Wider Powers	0.984139	0.943015	4084	4084	26	0	3577
12	Wider Endpoints	0.997234	0.933264	4110	4110	0	0	2577
14	Powers	0.977434	0.915155	4109	4109	1	0	3668
14	Endpoints	0.997161	0.927471	4110	4110	0	0	2647
14	Mixed	0.999989	0.997822	4110	4110	0	0	197
14	Wider Powers	0.984397	0.944628	4092	4092	18	0	3582
14	Wider Endpoints	0.997244	0.934263	4110	4110	0	0	2568

Table 28: NCI1: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	4110	-0.000432	-0.000892	-0.000050
edges	Mixed – Powers	4110	0.000177	0.000074	0.000305
edges	Mixed – Wider Powers	4109	-0.002258	-0.002858	-0.001696
edges	Wider Endpoints – Wider Powers	4109	-0.000518	-0.000965	-0.000165
nodes	Endpoints – Powers	4109	-0.002579	-0.003082	-0.002120
nodes	Mixed – Powers	4109	0.000132	0.000059	0.000222
nodes	Mixed – Wider Powers	4101	-0.001805	-0.002310	-0.001337
nodes	Wider Endpoints – Wider Powers	4101	-0.000680	-0.001084	-0.000361
pairs	Endpoints – Powers	4109	0.012316	0.007468	0.017259
pairs	Mixed – Powers	4109	0.082673	0.079010	0.086354
pairs	Mixed – Wider Powers	4092	0.053922	0.051052	0.056960
pairs	Wider Endpoints – Wider Powers	4092	-0.009385	-0.014170	-0.004636

J.4 NCI109

The report processes 4,127 graphs and 75 configurations.

Table 29: NCI109, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.993841	0.943802	4126	4127	0	0	2190
6	Endpoints	0.997706	0.977193	4126	4127	0	0	1042
6	Mixed	0.997901	0.978626	4126	4127	0	0	1005
6	Wider Powers	0.999601	0.995091	4126	4127	0	0	226
6	Wider Endpoints	0.999600	0.994786	4126	4127	0	0	239
8	Powers	0.997799	0.979101	4126	4127	0	0	964
8	Endpoints	0.997706	0.977193	4126	4127	0	0	1042
8	Mixed	0.998010	0.979800	4126	4127	0	0	946
8	Wider Powers	0.999891	0.998264	4126	4127	0	0	69
8	Wider Endpoints	0.999809	0.997181	4126	4127	0	0	119
10	Powers	0.999104	0.990537	4126	4127	0	0	425
10	Endpoints	0.999098	0.989859	4126	4127	0	0	454
10	Mixed	0.999232	0.990957	4126	4127	0	0	417
10	Wider Powers	0.999935	0.999106	4122	4123	4	0	35
10	Wider Endpoints	0.999892	0.998364	4126	4127	0	0	70
12	Powers	0.999601	0.995091	4126	4127	0	0	226
12	Endpoints	0.999600	0.994786	4126	4127	0	0	239
12	Mixed	0.999679	0.995337	4126	4127	0	0	222
12	Wider Powers	0.999946	0.999326	4122	4123	4	0	24
12	Wider Endpoints	0.999892	0.998364	4126	4127	0	0	70
14	Powers	0.999814	0.997392	4124	4125	2	0	117
14	Endpoints	0.999600	0.994786	4126	4127	0	0	239
14	Mixed	0.999864	0.997577	4124	4125	2	0	111
14	Wider Powers	0.999952	0.999476	4121	4122	5	0	19
14	Wider Endpoints	0.999926	0.998940	4126	4127	0	0	40

Table 30: NCI109, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.992718	0.929526	4126	4127	0	0	2428
6	Endpoints	0.998788	0.987500	4126	4127	0	0	563
6	Mixed	0.998939	0.988454	4126	4127	0	0	543
6	Wider Powers	0.999571	0.994671	4126	4127	0	0	249
6	Wider Endpoints	0.999751	0.996701	4126	4127	0	0	135
8	Powers	0.997416	0.975138	4126	4127	0	0	1151
8	Endpoints	0.998788	0.987500	4126	4127	0	0	563
8	Mixed	0.998977	0.988902	4126	4127	0	0	525
8	Wider Powers	0.999915	0.998649	4115	4116	11	0	73
8	Wider Endpoints	0.999854	0.998143	4126	4127	0	0	78
10	Powers	0.999005	0.989409	4126	4127	0	0	503
10	Endpoints	0.999478	0.993974	4126	4127	0	0	264
10	Mixed	0.999584	0.994688	4126	4127	0	0	244
10	Wider Powers	0.999954	0.999179	4118	4119	8	0	34
10	Wider Endpoints	0.999896	0.998782	4126	4127	0	0	43
12	Powers	0.999571	0.994671	4126	4127	0	0	249
12	Endpoints	0.999751	0.996701	4126	4127	0	0	135
12	Mixed	0.999829	0.997084	4126	4127	0	0	127
12	Wider Powers	0.999961	0.999354	4120	4121	6	0	28
12	Wider Endpoints	0.999896	0.998782	4126	4127	0	0	43
14	Powers	0.999812	0.997021	4126	4127	0	0	133
14	Endpoints	0.999751	0.996701	4126	4127	0	0	135
14	Mixed	0.999833	0.997135	4126	4127	0	0	124
14	Wider Powers	0.999965	0.999395	4125	4126	1	0	27
14	Wider Endpoints	0.999911	0.999134	4126	4127	0	0	29

Table 31: NCI109, pairs: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.854219	0.588310	4127	4127	0	0	4084
6	Endpoints	0.997266	0.896854	4127	4127	0	0	2944
6	Mixed	0.999781	0.974998	4127	4127	0	0	1292
6	Wider Powers	0.972612	0.895281	4127	4127	0	0	3741
6	Wider Endpoints	0.997646	0.927193	4127	4127	0	0	2612
8	Powers	0.928706	0.755752	4127	4127	0	0	3998
8	Endpoints	0.997266	0.896854	4127	4127	0	0	2944
8	Mixed	0.999884	0.984604	4127	4127	0	0	1103
8	Wider Powers	0.981833	0.931365	4106	4106	21	0	3640
8	Wider Endpoints	0.997683	0.931362	4127	4127	0	0	2565
10	Powers	0.959528	0.849151	4127	4127	0	0	3855
10	Endpoints	0.997553	0.918638	4127	4127	0	0	2708
10	Mixed	0.999970	0.994362	4127	4127	0	0	497
10	Wider Powers	0.984217	0.941374	4097	4097	30	0	3607
10	Wider Endpoints	0.997696	0.933422	4127	4127	0	0	2543
12	Powers	0.972612	0.895281	4127	4127	0	0	3741
12	Endpoints	0.997646	0.927193	4127	4127	0	0	2612
12	Mixed	0.999991	0.997597	4127	4127	0	0	258
12	Wider Powers	0.984709	0.944233	4106	4106	21	0	3608
12	Wider Endpoints	0.997696	0.933422	4127	4127	0	0	2543
14	Powers	0.978663	0.917965	4127	4127	0	0	3680
14	Endpoints	0.997646	0.927193	4127	4127	0	0	2612
14	Mixed	0.999994	0.998179	4127	4127	0	0	200
14	Wider Powers	0.984901	0.945500	4111	4111	16	0	3609
14	Wider Endpoints	0.997701	0.934444	4127	4127	0	0	2529

Table 32: NCI109: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	4127	-0.000320	-0.000693	-0.000029
edges	Mixed – Powers	4127	0.000114	0.000041	0.000202
edges	Mixed – Wider Powers	4126	-0.002213	-0.002732	-0.001731
edges	Wider Endpoints – Wider Powers	4126	-0.000250	-0.000577	-0.000005
nodes	Endpoints – Powers	4125	-0.002478	-0.002938	-0.002057
nodes	Mixed – Powers	4125	0.000185	0.000084	0.000314
nodes	Mixed – Wider Powers	4121	-0.001754	-0.002201	-0.001343
nodes	Wider Endpoints – Wider Powers	4122	-0.000388	-0.000684	-0.000148
pairs	Endpoints – Powers	4127	0.009228	0.004397	0.013945
pairs	Mixed – Powers	4127	0.080214	0.076613	0.083838
pairs	Mixed – Wider Powers	4111	0.053187	0.050400	0.056096
pairs	Wider Endpoints – Wider Powers	4111	-0.010181	-0.014962	-0.005476

J.5 PROTEINS

The report processes 1,113 graphs and 50 configurations.

Table 33: PROTEINS, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.998961	0.987262	1101	1113	0	0	172
6	Endpoints	0.998084	0.992324	1101	1113	0	0	109
6	Mixed	0.999519	0.993925	1101	1113	0	0	102
6	Wider Powers	0.999596	0.997992	1101	1113	0	0	35
6	Wider Endpoints	0.998348	0.997041	1101	1113	0	0	41
8	Powers	0.999400	0.994089	1101	1113	0	0	95
8	Endpoints	0.998084	0.992324	1101	1113	0	0	109
8	Mixed	0.999535	0.994397	1101	1113	0	0	93
8	Wider Powers	0.999631	0.998922	1098	1110	3	0	20
8	Wider Endpoints	0.998390	0.997850	1101	1113	0	0	22
10	Powers	0.999536	0.996808	1100	1112	1	0	61
10	Endpoints	0.998248	0.995311	1101	1113	0	0	71
10	Mixed	0.999671	0.997064	1101	1113	0	0	60
10	Wider Powers	0.999639	0.999065	1100	1112	1	0	16
10	Wider Endpoints	0.998402	0.998203	1100	1112	1	0	16
12	Powers	0.999596	0.997992	1101	1113	0	0	35
12	Endpoints	0.998348	0.997041	1101	1113	0	0	41
12	Mixed	0.999749	0.998564	1101	1113	0	0	32
12	Wider Powers	0.999650	0.999242	1101	1113	0	0	12
12	Wider Endpoints	0.998402	0.998203	1100	1112	1	0	16
14	Powers	0.999622	0.998593	1100	1112	1	0	27
14	Endpoints	0.998348	0.997041	1101	1113	0	0	41
14	Mixed	0.999759	0.998933	1101	1113	0	0	26
14	Wider Powers	0.999656	0.999356	1100	1112	1	0	8
14	Wider Endpoints	0.998409	0.998373	1100	1112	1	0	8

Table 34: PROTEINS, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.999204	0.982437	1101	1113	0	0	243
6	Endpoints	0.997457	0.990299	1101	1113	0	0	133
6	Mixed	0.999855	0.995422	1101	1113	0	0	92
6	Wider Powers	0.999941	0.998164	1098	1110	3	0	47
6	Wider Endpoints	0.997627	0.994403	1101	1113	0	0	76
8	Powers	0.999718	0.992514	1101	1113	0	0	132
8	Endpoints	0.997457	0.990299	1101	1113	0	0	133
8	Mixed	0.999856	0.995521	1101	1113	0	0	88
8	Wider Powers	0.999967	0.999155	1099	1111	2	0	20
8	Wider Endpoints	0.997646	0.995094	1101	1113	0	0	61
10	Powers	0.999876	0.996294	1099	1111	2	0	73
10	Endpoints	0.997559	0.992836	1101	1113	0	0	101
10	Mixed	0.999931	0.997751	1101	1113	0	0	57
10	Wider Powers	0.999978	0.999435	1099	1111	2	0	16
10	Wider Endpoints	0.997655	0.995427	1100	1112	1	0	52
12	Powers	0.999941	0.998164	1098	1110	3	0	47
12	Endpoints	0.997627	0.994403	1101	1113	0	0	76
12	Mixed	0.999978	0.999079	1101	1113	0	0	31
12	Wider Powers	0.999987	0.999650	1099	1111	2	0	11
12	Wider Endpoints	0.997655	0.995427	1100	1112	1	0	52
14	Powers	0.999958	0.998838	1099	1111	2	0	29
14	Endpoints	0.997627	0.994403	1101	1113	0	0	76
14	Mixed	0.999979	0.999174	1101	1113	0	0	29
14	Wider Powers	0.999989	0.999729	1101	1113	0	0	8
14	Wider Endpoints	0.997659	0.995531	1101	1113	0	0	49

Table 35: PROTEINS: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	1111	-0.004432	-0.006818	-0.002425
edges	Mixed – Powers	1111	0.000346	0.000094	0.000676
edges	Mixed – Wider Powers	1113	-0.000555	-0.000943	-0.000162
edges	Wider Endpoints – Wider Powers	1113	-0.004197	-0.006557	-0.002211
nodes	Endpoints – Powers	1112	-0.001528	-0.003068	-0.000230
nodes	Mixed – Powers	1112	0.000366	0.000022	0.000870
nodes	Mixed – Wider Powers	1112	-0.000393	-0.000923	0.000201
nodes	Wider Endpoints – Wider Powers	1111	-0.001006	-0.002567	0.000252

J.6 MOLHIV

The report processes 41,127 graphs and 50 configurations.

Table 36: molhiv, nodes: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.993101	0.950440	41103	41127	0	0	17595
6	Endpoints	0.997169	0.978024	41103	41127	0	0	8470
6	Mixed	0.997480	0.979900	41103	41127	0	0	7964
6	Wider Powers	0.999421	0.994902	41103	41127	0	0	1969
6	Wider Endpoints	0.999409	0.994741	41103	41127	0	0	2066
8	Powers	0.997281	0.979849	41103	41127	0	0	7724
8	Endpoints	0.997169	0.978024	41103	41127	0	0	8470
8	Mixed	0.997580	0.980874	41103	41127	0	0	7507
8	Wider Powers	0.999838	0.998392	41063	41087	40	0	647
8	Wider Endpoints	0.999679	0.997038	41103	41127	0	0	1205
10	Powers	0.998794	0.990368	41103	41127	0	0	3762
10	Endpoints	0.998780	0.989896	41103	41127	0	0	3984
10	Mixed	0.998955	0.990971	41103	41127	0	0	3631
10	Wider Powers	0.999908	0.999016	41073	41097	30	0	351
10	Wider Endpoints	0.999807	0.998245	41103	41127	0	0	682
12	Powers	0.999421	0.994902	41103	41127	0	0	1969
12	Endpoints	0.999409	0.994741	41103	41127	0	0	2066
12	Mixed	0.999510	0.995273	41103	41127	0	0	1904
12	Wider Powers	0.999934	0.999349	41075	41099	28	0	224
12	Wider Endpoints	0.999807	0.998245	41103	41127	0	0	682
14	Powers	0.999699	0.997087	41102	41126	1	0	1200
14	Endpoints	0.999409	0.994741	41103	41127	0	0	2066
14	Mixed	0.999753	0.997366	41102	41126	1	0	1124
14	Wider Powers	0.999949	0.999503	41068	41092	35	0	173
14	Wider Endpoints	0.999862	0.998809	41103	41127	0	0	423

Table 37: molhiv, edges: complete width scan.

d	Route	PR	RC	N_R	N_C	U	I	M
6	Powers	0.991391	0.936027	41087	41127	0	0	21257
6	Endpoints	0.998324	0.987387	41087	41127	0	0	4726
6	Mixed	0.998584	0.988894	41087	41127	0	0	4305
6	Wider Powers	0.999253	0.994342	41087	41127	0	0	2111
6	Wider Endpoints	0.999584	0.996461	41087	41127	0	0	1335
8	Powers	0.996703	0.976646	41087	41127	0	0	8967
8	Endpoints	0.998324	0.987387	41087	41127	0	0	4726
8	Mixed	0.998626	0.989331	41087	41127	0	0	4122
8	Wider Powers	0.999715	0.997981	41072	41112	15	0	769
8	Wider Endpoints	0.999752	0.997928	41087	41127	0	0	762
10	Powers	0.998530	0.989392	41087	41127	0	0	4040
10	Endpoints	0.999213	0.993661	41087	41127	0	0	2335
10	Mixed	0.999370	0.994527	41087	41127	0	0	2069
10	Wider Powers	0.999810	0.998905	41044	41084	43	0	410
10	Wider Endpoints	0.999822	0.998625	41087	41127	0	0	460
12	Powers	0.999253	0.994342	41087	41127	0	0	2111
12	Endpoints	0.999584	0.996461	41087	41127	0	0	1335
12	Mixed	0.999691	0.996948	41087	41127	0	0	1213
12	Wider Powers	0.999840	0.999214	41045	41085	42	0	278
12	Wider Endpoints	0.999822	0.998625	41087	41127	0	0	460
14	Powers	0.999567	0.996678	41087	41127	0	0	1301
14	Endpoints	0.999584	0.996461	41087	41127	0	0	1335
14	Mixed	0.999697	0.997027	41087	41127	0	0	1182
14	Wider Powers	0.999853	0.999321	41054	41094	33	0	235
14	Wider Endpoints	0.999861	0.999002	41087	41127	0	0	303

Table 38: molhiv: paired RC differences at $d = 14$.

Domain	Contrast	N	Mean Δ	CI low	CI high
edges	Endpoints – Powers	41127	-0.000216	-0.000345	-0.000091
edges	Mixed – Powers	41127	0.000349	0.000280	0.000423
edges	Mixed – Wider Powers	41094	-0.002034	-0.002220	-0.001851
edges	Wider Endpoints – Wider Powers	41094	-0.000173	-0.000294	-0.000058
nodes	Endpoints – Powers	41126	-0.002343	-0.002489	-0.002202
nodes	Mixed – Powers	41126	0.000279	0.000232	0.000331
nodes	Mixed – Wider Powers	41091	-0.001857	-0.002011	-0.001705
nodes	Wider Endpoints – Wider Powers	41092	-0.000492	-0.000593	-0.000395

K Dataset Coverage and Additional Capacity Results

Dataset sources and graph versions are given in Appendix E.2. ZINC subset, QM9, and the six datasets in Appendix J use channel-prefix width $d = 14$. Other rows use the supplied 2026-09-24 snapshot at reported $Q = 14$. PROTEINS covers all 1,113 graphs on nodes and edges. Historical degree labels are not equated with current channel prefixes.

Dataset graph counts describe processing scope, not the route-specific eligible populations used to average Representative Capacity.

Table 39: Dataset coverage. Counts are graphs per dataset, not sums over domains. A dagger marks the historical distance-gated protocol ($L = 5$); its results are not pooled with ungated capacity.

Dataset	Reported modes	Graphs
MNIST	nodes	70,000
CIFAR10	nodes	60,000
CLUSTER†	nodes, nodes+edges	12,000
PATTERN†	nodes, nodes+edges	14,000
VOC†	nodes, nodes+edges	11,355
PPA†	nodes, edges	158,100
ZINC subset	nodes, edges, pairs	12,000
ZINC full	nodes, edges, pairs	249,456
QM9	nodes, edges, pairs	130,831
molhiv	nodes, nodes+edges	41,127
pcba†	nodes, nodes+edges	437,929
MUTAG	nodes, edges, pairs	188
NCI1	nodes, edges, pairs	4,110
NCI109	nodes, edges, pairs	4,127
ENZYMES	nodes, edges, pairs	600
PROTEINS	nodes, edges	1,113
Peptides-functional	nodes, nodes+edges	15,535

Table 40: Node Representative Capacity at the indicated width or historical $Q = 14$. A dagger marks distance-gated results. Detailed appendices give eligible counts for the width scans; historical counts are unavailable. Distance-gated PCBA nodes are discussed below.

Dataset	Powers	Endpoints	Mixed
MNIST	1.000	1.000	1.000
CIFAR10	1.000	1.000	1.000
CLUSTER†	1.000	1.000	1.000
PATTERN†	1.000	1.000	1.000
PPA†	0.999998	0.999997	0.999998
VOC†	0.99946	0.99903	0.99938
ZINC subset ($d = 14$)	0.999967	0.999343	0.999991
ZINC full	0.99992	0.99927	0.99996
QM9 ($d = 14$)	0.999820	0.999933	0.999996
molhiv ($d = 14$)	0.997087	0.994741	0.997366
NCI1 ($d = 14$)	0.997110	0.994473	0.997242
NCI109 ($d = 14$)	0.997392	0.994786	0.997577
ENZYMES ($d = 14$)	0.997855	0.997404	0.998271
PROTEINS ($d = 14$)	0.998593	0.997041	0.998933
Peptides-functional	0.93589	0.89320	0.93589
MUTAG ($d = 14$)	0.998550	0.996333	0.998550

Aggregation and monotonicity. The current reporting implementation computes route-specific means over eligible graphs, requiring reliability and exact MIS for the primary capacity summary. Paired comparisons use a shared eligible population. The older summaries do not include those eligible populations. The ZINC subset rerun has 12,000 common reliable graphs and exact MIS solutions for every reported configuration. QM9 valid populations and paired estimates are given

Table 41: Edge Representative Capacity at the indicated width or historical $Q = 14$. A dagger marks runs with the recorded distance gate; detailed appendices give eligible counts for the width scans.

Dataset	Powers	Endpoints	Mixed
CLUSTER†	1.000	1.000	1.000
PATTERN†	1.000	1.000	1.000
PPA†	0.999988	0.999978	0.999995
pcba†	0.999499	0.999208	0.999555
VOC†	0.99967	0.99903	0.99963
ZINC subset ($d = 14$)	0.999900	0.999652	0.999970
ZINC full	0.99984	0.99959	0.99994
QM9 ($d = 14$)	0.999555	0.999870	0.999998
molhiv ($d = 14$)	0.996678	0.996461	0.997027
NCI1 ($d = 14$)	0.996697	0.996264	0.996873
NCI109 ($d = 14$)	0.997021	0.996701	0.997135
ENZYMES ($d = 14$)	0.997946	0.995893	0.997969
PROTEINS ($d = 14$)	0.998838	0.994403	0.999174
Peptides-functional	0.93442	0.93393	0.93447
MUTAG ($d = 14$)	0.999106	0.998768	0.999106

Table 42: Pair Representative Capacity. QM9 uses all unordered distinct pairs from the width scan (Appendix I).

Dataset	Domain	Powers	Endpoints	Mixed
ZINC subset ($d = 14$)	pairs	0.975404	0.988718	0.999987
ZINC full	pairs	0.97505	0.98858	0.99997
QM9 ($d = 14$)	pairs	0.910424	0.975929	0.999986
ENZYMES ($d = 14$)	pairs	0.966190	0.988143	0.997936
MUTAG ($d = 14$)	pairs	0.980234	0.929131	0.999572
NCI1 ($d = 14$)	pairs	0.915155	0.927471	0.997822
NCI109 ($d = 14$)	pairs	0.917965	0.927193	0.998179

in Appendix I. In particular, the VOC Mixed means are slightly below Powers despite the latter's features being retained. Different populations can produce this pattern, but the summary alone does not establish its cause. The per-graph eligibility and numerical masks are required to resolve the discrepancy. We therefore report these as descriptive means rather than claim that Mixed uniformly improves capacity on identical instances.

Distance-gated results. CLUSTER, PATTERN, VOC, PPA, and pcba use the historical distance gate with $L = 5$. For nodes, two objects are compared only when their shortest-path distance is at most L. For two pair objects $\{u, v\}$ and $\{u', v'\}$, the minimum of all six endpoint distances is used, including the distances within each object. In particular, any two existing edges pass this gate when $L \geq 1$. Beyond-gate comparisons are removed from the Resolution denominator and from the similarity graph used for MIS.

PCBA node RC is 1.01193, 1.01128, and 1.01194 for Powers, Endpoints, and Mixed. Multiple objects from one orbit can be selected when their comparison is omitted, so MIS divided by the full-graph orbit count can exceed one. This is not the ungated capacity metric. We neither clip these values nor interpret them as exceeding the orbit ceiling.

Omitted datasets. COCO, code2, and MalNetTiny have no reported measurements in this study. Peptides-structural is omitted because it shares graph topologies with Peptides-functional. These conditions are not counted as measured.

K.1 Matched-Encoding Comparisons

The completed width controls for ZINC subset, QM9, and six additional datasets appear in Appendices H, I, and J. They use channel prefixes at $d = 6, 8, 10, 12, 14$, with explicit object domains and eligible populations. In these structural experiments, Mixed versus Wider Powers matches $2d$ channels per orientation, and Powers versus Endpoints matches d. These historical width conventions remain distinct from the newer GNNPlus equal-total-width protocol.

Paired capacity differences use common eligible graphs. Graph-bootstrap intervals describe variation across evaluated graphs, not training seeds. Numerical reliability exclusions, inexact MIS solves, and unresolved object comparisons are reported separately; a high mean does not certify full orbit recovery. Distance-gated results remain separate from ungated capacity averages. Reported aggregate run costs are not independent preprocessing benchmarks for individual routes.

K.2 Scope of the GNNPlus Downstream Observation

Real-task prediction is discussed only in the GNNPlus setting (Luo et al., 2025). The completed encoding comparisons are reported to remain close to the published GNNPlus results, without a consistent advantage across encoding families. Task scores and seed-level uncertainty are not supplied here, so this remains a qualitative observation. Published GNNPlus scores are attributed to that work and are not substituted for measurements from our runs. All ZINC, QM9, and TU tables in this paper report within-graph object discrimination, not supervised prediction accuracy or regression error.

The current implementation supports equal total width 20: Powers uses $(D^{-1}A)^k$ channels; Endpoints uses products of entries of $h_a = (D^{-1/2}AD^{-1/2})^a\mathbf{1}$; Mixed selects ten channels from each family. Implementation support does not establish that historical runs used this protocol. The object-discrimination tables retain their symmetric-normalization and branch-width settings. They therefore do not supply measurements of every downstream input configuration.

K.3 Interpretation of Structural and Predictive Changes

Structural capacity measures distinctions on the chosen objects, not predictive relevance. A pair-domain gain is not an edge-domain gain. Even a matching structural improvement need not be useful for the target or exploited during training. Small measured node/edge increments are consistent with limited downstream gains: Peptides-functional Powers and Mixed capacities differ by only 0.000001 on nodes and 0.000052 on edges in the historical summaries. This is a possible explanation, not a causal result or evidence that stronger architectures are unnecessary.