Beyond Dense Tensors: Exact Evaluation of Spectral Hypergraph Layers

Anonymous authors
Paper under double-blind review

ABSTRACT

Spectral hypergraph neural networks encode each hyperedge in a dense adjacency tensor, but their storage grows as the number of vertices raised to the size of the largest hyperedge, which makes them prohibitively expensive even on modest hypergraphs. We show that this cost is avoidable for the T-Spectral layer as released, whose readout sums all output slices of the t-product. We eliminate the tensor intermediates algebraically and evaluate the layer directly from node –hyperedge incidence data, preserving its output, normalisation and treatment of repeated hyperedges. After preprocessing, propagation takes time linear in the number of vertices and incidence entries for a fixed feature width, and neither the adjacency tensor nor its pairwise expansion is ever built. The same derivation reveals an expressivity limit already present in the original layer: on a labelled Pasch configuration, two differently labelled hypergraphs induce identical effective operators, which forces chance-level accuracy on a balanced task whose labels are recoverable from the incidence structure. Exact evaluation inherits this limit. Together, these results remove the dense-tensor barrier to evaluating the layer on larger hypergraphs and identify the structural distinctions that require a richer architecture.

1 INTRODUCTION

The sum readout of the released T-Spectral layer makes dense tensor computation unnecessary and limits which hypergraph structure the layer can use. We derive both consequences from the same identity. A hypergraph stores group relations as hyperedges; tensor methods encode these relations in an adjacency tensor and use the t-product for spectral filtering (Banerjee et al., 2017; Peña-Peña et al., 2023; Wang et al., 2025). For the released layer, summing the output slices reduces this computation to an operator evaluated directly from incidence data.

The obstacle is size. The adjacency tensor has N^M entries, where N is the number of nodes and M the size of the largest hyperedge, and M is set by the data rather than chosen. Wang et al. (2025) note this $O(N^M)$ memory cost; to evaluate their tensor layers at all, they downsampled five academic-network hypergraphs. The largest of these has 89 nodes, and their dense tensors still take 100 to 500 MB each. At full size the tensor cannot be stored: for the full-size House committee hypergraph of the reference implementation's dataset archive, whose largest hyperedge has 81 members, it would need about 7×10^{252} bytes (Table 5).

The released layer does not require the tensor. The t-product convolves along the last $M-2$ indices of the tensor, its *tube axes*, and each position on them holds one $N \times N$ adjacency slice. The layer sums the output slices of this convolution, and the sum of a circular convolution is a product of sums:

$$\textstyle\sum_k (f * g)_k \;=\; \big(\sum_k f_k\big)\big(\sum_k g_k\big).$$

Applied along each tube axis, this identity reduces the whole layer to one $N \times N$ matrix $\bar{A}$, the tensor summed over its tube axes, applied to the features and scaled by one number per channel: the multiplicative interaction survives only as this global scale. $\bar{A}$ can in turn be read off the incidence matrix by counting and applied through three sparse products, so neither the tensor nor $\bar{A}$ itself need ever be formed (Figure 1). Our contributions are:

1. **Exact evaluation from incidence data** (§3–§4, §6). We evaluate the released layer without forming its adjacency tensor or its pairwise expansion, while preserving normalisation and

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

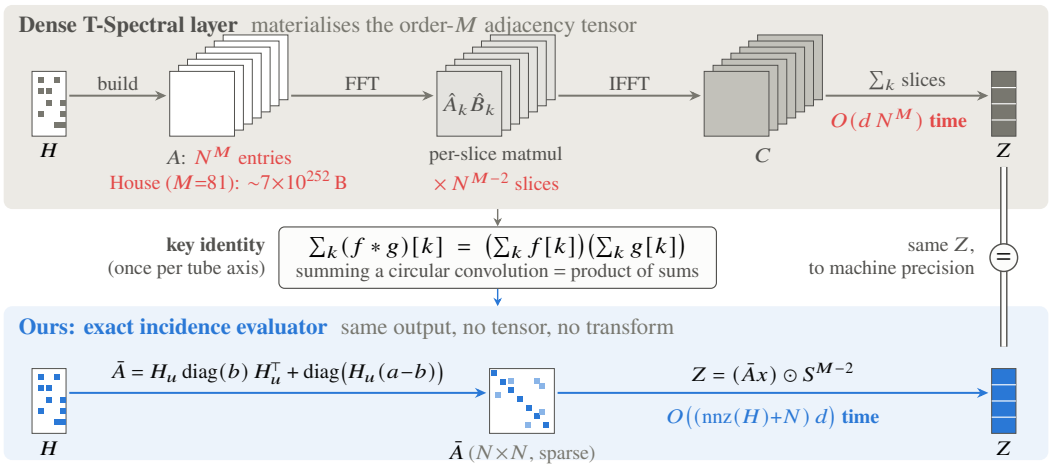


Figure 1: Exact evaluation and the expressivity limit share the same reduction. Both routes use $\bar{A}$ and S^{M-2}, where S holds the channel sums. The dense route forms the tensor and applies the t-product; the incidence route applies $\bar{A}$ through sparse products without forming it. In the image, $\widehat{B}$ denotes the transformed interaction signal ($\hat{X}$ in the text). Output parity is checked in §4.

repeated-hyperedge semantics. The effective operator is the all-ones case of the tensor-times-same-vector contraction (TTSV2) of Aksoy et al. (2024); the contribution is the exact reduction of the released layer to this contraction. The evaluator matches dense evaluation to machine precision, is over a million times faster per training step, and runs the layer on full-size hypergraphs with up to 88,860 nodes (Li et al., 2026).

2. **An expressivity limit inherited by exact evaluation** (§3.4, §5). The same reduction identifies what the released layer sees of the hypergraph: how strongly each *pair* of nodes co-occurs, not which larger groups they form (Theorem 3). On a labelled Pasch configuration, two labellings with the same pairs but different triangles, it is held at chance (50%), while set-based networks reach 100%. The released layer already has this limit; no other fixed weighting of the output slices respects relabelling (Theorem 2: proved at $M = 3$, conjectured for general M, enumerated exactly at $M \in \{4, 5\}$, $N \leq 6$).
3. **A closed form for the printed (symmetrised) layer** (§5). We derive the exact closed form of the printed (symmetrised) layer of Wang et al. (2025) and show why it retains higher-order information: it is equivariant, does not collapse, and scores 100% on the Pasch task. It avoids the dense array, though not the tensor's nonzero tuples.

2 Background: the T-Spectral HyperGNN

The T-Spectral layer takes the t-product of an order-M adjacency tensor and a feature interaction signal, both with $M-2$ tube axes of length N. Three objects share its name. *The released layer*, in the reference implementation, is the unsymmetrised t-product with the sum readout defined below; we analyse and train it. *The printed (symmetrised) layer* of Wang et al. (2025, §III-B–C) pads each tube axis symmetrically before the t-product and removes the padding before the readout; the reference implementation defines this padding but never calls it (Appendix H). *T-Spectral (exact)* is our evaluator of the released layer and, in the tables, the model trained with it.

Hypergraphs and the order M. A hypergraph on nodes $V = \{1, \ldots, N\}$ has hyperedges $e \subseteq V$ and incidence matrix $H \in \{0,1\}^{N\times|E|}$, $H_{ie} = 1$ iff $i \in e$. Its order is

$$M = \max_{e\in E} |e|, \tag{1}$$

set by the data. Our running example has $N = 3$ and two hyperedges, $e_1 = \{1, 2\}$ and $e_2 = \{1, 2, 3\}$, so $M = 3$.

The adjacency tensor. Following Wang et al. (2025, Def. 1), building on Banerjee et al. (2017), a hyperedge e of size $c \le M$ writes into every length-M index tuple that uses each node of e at least once and no other node. There are

$$L_{M,c} \;=\; c!\,S(M,c) \;=\; \sum_{r=0}^{c}(-1)^r\binom{c}{r}(c-r)^M \tag{2}$$

such tuples: the number of ways to fill M slots with c nodes so that none is left out. Here $S(M,c)$ is the Stirling number of the second kind, the number of ways to split M slots into c non-empty groups. Each tuple receives the value $c/L_{M,c}$. A hyperedge smaller than M is thus padded by repeating its own nodes, and A is symmetric in all M indices. In the example, e_1 writes $2/6 = 1/3$ into the six tuples over $\{1,2\}$ that use both nodes, and e_2 writes $3/6 = 1/2$ into the six orderings of $(1,2,3)$: 12 nonzeros out of $3^3 = 27$ entries. The reference implementation uses A by default and optionally the normalised tensor $A^{\text{norm}} = D^{\dagger}A$, $D = \operatorname{diag}(H\mathbf{1})$ ($D^{\dagger}$ inverts nonzero degrees, so $D^{\dagger} = D^{-1}$ when no node is isolated), dividing each entry by the degree of its first-index node; we cover both.

The interaction signal and the t-product. Features enter through one linear layer with a ReLU, $x \in \mathbb{R}^{N\times d}_{\ge 0}$, and each channel c forms the $(M-1)$-fold outer product $X[j,c,k_1,\dots,k_T] = x_{jc}\,x_{k_1c}\cdots x_{k_Tc}$ with $T := M-2$ *tube axes* (Wang et al., 2025, eq. 6), modelling multiplicative interactions among nodes. Writing $A^{[k]}$ and $X^{[k]}$ for the slices at tube position $k \in [N]^T$, the t-product (Kilmer and Martin, 2011; Kilmer et al., 2013) convolves circularly along every tube axis,

$$C^{[k]} \;=\; \sum_{a\in[N]^T} A^{[a]}\,X^{[k-a]} \qquad \text{(indices mod } N \text{ on each axis)}, \tag{3}$$

computed by FFTs along the tube axes, one matrix product per frequency, and an inverse FFT. The readout sums all slices (Wang et al., 2025, eq. 15), $Z = \sum_k C^{[k]}W$, with W mixing channels. Wang et al. (2025) use a single layer, since the t-product mixes information from all nodes in one step. The $O(dN^M)$ time of the t-product motivated their T-Spatial convolution.

Slice pairings. The released layer pairs every tensor slice with every signal slice; T-Spatial uses matched slices (Wang et al., 2025, eqs. 16–17). Table 12 (Appendix H) gives the readouts side by side.

3 Exact evaluation without the tensor

The released layer can be computed exactly without forming its N^M tensor, in two steps: the readout removes the tube axes (§3.1), and the remaining $N\times N$ matrix is read off the incidence matrix (§3.2).

3.1 The readout collapses the tube axes

Start with one tube axis of length 2, $f = (1,2)$ and $g = (3,4)$. Their circular convolution is $f * g = (1\cdot 3 + 2\cdot 4,\;\; 1\cdot 4 + 2\cdot 3) = (11,10)$, and its sum is $21 = (1+2)(3+4)$. This holds in general: as the output position k runs over all positions, so does $k-a$, and every product $f_a g_b$ appears in the sum exactly once. In Fourier terms, the sum of a signal is its zero-frequency coefficient, and convolution multiplies coefficients frequency by frequency, axis by axis.

Proposition 1 (Collapse). *Let $M \ge 2$, $T = M-2$, let A be any tensor with T tube axes (no symmetry is needed), and let $x \in \mathbb{R}^{N\times d}$. Write $\bar{A}[i,j] := \sum_{a\in[N]^T} A[i,j,a]$ for the tube-summed adjacency and $S := x^{\top}\mathbf{1} \in \mathbb{R}^d$ for the channel sums. Then the sum readout of the layer of §2 is*

$$\sum_k C^{[k]} \;=\; \bar{A}\,x\,\operatorname{diag}(S)^{M-2}, \qquad \textit{so} \qquad Z = \bar{A}\,x\,\operatorname{diag}(S)^{M-2}\,W. \tag{4}$$

Proof. The t-product multiplies frequency by frequency, $\hat{C}^{[\omega]} = \hat{A}^{[\omega]}\hat{X}^{[\omega]}$, and at $\omega = 0$ the (unnormalised) DFT is a plain sum. So $\sum_k C^{[k]} = \hat{C}^{[0]} = \bar{A}\sum_l X^{[l]}$, and

$$\textstyle\sum_l X[j,c,l] = x_{jc}\prod_{r=1}^{T}\sum_{l_r} x_{l_rc} = x_{jc}\,S_c^T. \qquad \square$$

The layer therefore sees the hypergraph only through the $N \times N$ matrix $\bar{A}$ (a 3×3 matrix in the example, instead of 27 tensor entries), and the other $M-2$ factors of the interaction signal only through the d channel sums S. The multiplicative interactions that motivate the layer thus reduce to one global factor S_c^{M-2} per channel, and node-to-node propagation is a single step of $\bar{A}$, a weighted clique expansion of the hypergraph, also called its two-section. Since W commutes with the sum and the degree normaliser acts on the first index, normalising gives $\bar{A}^{\mathrm{norm}} = D^{\dagger}\bar{A}$.

3.2 Reading $\bar{A}$ off the incidence matrix

Each entry of $\bar{A}$ is still a sum over N^{M-2} tube positions; counting avoids that sum. In the example, $\bar{A}[1,2] = \sum_k A[1,2,k]$. From e_1, both $k=1$ and $k=2$ complete a valid tuple, contributing $2 \times \frac{1}{3}$; from e_2, only $k=3$ does, contributing $\frac{1}{2}$. So $\bar{A}[1,2] = \frac{7}{6}$. On the diagonal, $\bar{A}[1,1]$ receives $\frac{1}{3}$ from $(1,1,2)$ but nothing from e_2: once two slots hold node 1, one slot is left for nodes 2 and 3. Altogether

$$\bar{A} = \begin{pmatrix} \frac{1}{3} & \frac{7}{6} & \frac{1}{2} \\ \frac{7}{6} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}.$$

In general, a hyperedge e of size c adds $c/L_{M,c}$ to $\bar{A}[i,j]$ for every way of filling the remaining $M-2$ slots so that all of e appears. That count depends only on whether $i=j$ (the free slots must cover the other $c-1$ nodes) or $i \neq j$ (they must cover $c-2$). Each hyperedge therefore contributes one *self weight* $a_{M,c}$ to each member and one *pair weight* $b_{M,c}$ to each ordered pair of distinct members.

The reference builder writes entries by assignment, so a repeated node-set is written once and $\bar{A}$ depends only on the distinct node-sets, while its degree normaliser counts every copy. Our construction preserves both conventions and matches the dense tensor entry for entry (Appendix E.5).

Theorem 1 ($\bar{A}$ from the incidence matrix)**.** *Let $\mathcal{E}_u$ be the distinct node-sets among the hyperedges and $H_u \in \{0,1\}^{N \times |\mathcal{E}_u|}$ their incidence matrix, with columns h_e. With $S(\cdot,\cdot)$ the Stirling numbers of the second kind, let*

$$a_{M,c} = \frac{S(M-1,c)}{S(M,c)} = \frac{L_{M-1,c}}{L_{M,c}}, \qquad b_{M,c} = \frac{1 - a_{M,c}}{c-1} \quad (c \geq 2), \qquad a_{M,1} = 1,\ b_{M,1} = 0. \tag{5}$$

Then, with $a, b \in \mathbb{R}^{|\mathcal{E}_u|}$ holding $a_{M,|e|}$ and $b_{M,|e|}$,

$$\bar{A} = \textstyle\sum_{e \in \mathcal{E}_u} \left[b_{M,|e|}\, h_e h_e^{\top} + \left(a_{M,|e|} - b_{M,|e|}\right) \mathrm{diag}(h_e) \right] \tag{6}$$

$$= H_u \,\mathrm{diag}(b)\, H_u^{\top} + \mathrm{diag}\big(H_u(a-b)\big), \tag{7}$$

and the normalised layer uses $\bar{A}^{\mathrm{norm}} = D^{\dagger}\bar{A}$ with $D = \mathrm{diag}(H\mathbf{1})$, the degree counted in the full incidence matrix ($D^{\dagger}$ leaves isolated nodes at zero).

Proof: Appendix C.

3.3 The algorithm and its cost

Each evaluation is three sparse products (Theorem 1): gather to hyperedges, scatter, fix the diagonal.

T-Spectral (exact): the released layer from incidence data. *Input:* sparse H, features X_0, parameters W_0, b_0, W.
Once per hypergraph. (1) Deduplicate the node-sets of H (hashing each support) to obtain H_u; record $D = \mathrm{diag}(H\mathbf{1})$ (normalised setting only). (2) For each hyperedge size c that occurs, compute $a_{M,c}, b_{M,c}$ from (5) in exact integer arithmetic and round once. (3) Store the vectors b and $a-b$ over $\mathcal{E}_u$.
Every evaluation. (4) $x = \mathrm{ReLU}(X_0 W_0 + \mathbf{1} b_0^{\top})$, $S = x^{\top}\mathbf{1}$.
(5) $U = D^{\dagger}\left[H_u\big(b \odot (H_u^{\top} x)\big) + (H_u(a-b)) \odot x\right]$, where $b\odot$ scales row e by b_e; $D^{\dagger}$ is omitted in the unnormalised setting. (6) $Z = U\,\mathrm{diag}(S)^{M-2}\,W$, with the factor in the log domain or absorbed into W (Remark 1); then dropout and log-softmax.

The one-time step is $O(\mathrm{nnz}(H))$ expected and each evaluation $O((\mathrm{nnz}(H_u) + N)\,d)$, against N^M stored entries and $O(dN^M)$ time for the dense route (Table 4, Appendix E.2).

Remark 1 (What the one-layer model computes)**.** With fixed input features X_0 (the transductive setting), S is fixed once the encoder parameters are fixed and $\mathrm{diag}(S)^{M-2}W =: W_1$ is a single matrix, so the released model (the released layer with its encoder) realises exactly

$$X_0 \longmapsto \widetilde{A}\,\mathrm{ReLU}(X_0W_0 + \mathbf{1}b_0^\top)\,W_1, \qquad \widetilde{A} \in \{\bar{A},\ \bar{A}^{\mathrm{norm}}\}, \tag{8}$$

and every W_1 is reached (for the nonnegative post-ReLU features; Remark 3): the same encoder–propagate–decode structure as SGC and APPNP (Wu et al., 2019; Gasteiger et al., 2019), here derived from a tensor layer. Across graphs, S and M change with the input (§8). We train T-Spectral (exact) in this form with W_1 free, keeping the hypothesis class but not the training dynamics (§8); on DHG-Bench a normalisation may precede W_1 (Appendix I; search space in Appendix J).

Numerical evaluation. The unabsorbed factor can also be evaluated in the log domain (Appendix G.3). The trained T-Spectral (exact) models use the absorbed form above; training through the log-domain factor remains untested.

3.4 Equivariance of fixed readouts

Individual output slices may depend on node ordering (Wang et al., 2025). Theorem 2 asks which fixed readouts respect relabelling.

Definition 1 (Index-keyed readout)**.** A tube readout $Z_w = \sum_k w_k\, C^{[k]}$ is *index-keyed* if the weights $w \in \mathbb{R}^{N^T}$ are fixed parameters attached to tube positions, independent of the features and not permuted when the nodes are relabelled.

Theorem 2 (The sum is the only equivariant fixed readout, $M = 3$)**.** *Let $M = 3$, $N \geq 5$, and let A be the unnormalised tensor of §2. For an index-keyed weighting w, and up to weightings with $Z_w \equiv 0$, the map from a hypergraph and its features to Z_w is equivariant under all node permutations if and only if w is constant.*

Proof: Appendix B. For general M we conjecture the same with w symmetrised over the tube axes (which Z_w cannot see); exact enumeration confirms it at $M \in \{4, 5\}$ for $N \leq 6$ (Appendix D.2). Readouts that change the product lie outside the theorem: T-Spatial uses a slice-matched product, and the printed (symmetrised) layer pads the inputs of the t-product (§5).

4 Parity and cost

We call the reference implementation's dataset collection *the archive*. Its five downsampled academic networks, where the dense route runs, are the *reference hypergraphs*; it also holds two full-size ones, House and smallWalmart, beyond its reach (Table 5). We compare the exact evaluator, with the factor S^{M-2} kept, against a dense oracle built from the definition (N^M tensor, FFTs, one matrix product per slice, inverse FFT, slice sum).

Numerical agreement. On the five reference hypergraphs we compare outputs, loss, and feature and weight gradients. In double precision the two routes agree to machine precision (relative differences below 10^{-15}). Against the reference implementation, which computes in single-precision complex arithmetic, they agree to its rounding level, about 10^{-7}. All 750 comparisons pass (Table 10; normalised setting: Appendix G.1), as do 3,750 more on 25 small hypergraphs cut from House and smallWalmart, adding optimiser trajectories and relabelling (Table 11).

Runtime and memory. On the reference hypergraphs (Figure 2), one dense training step (the reference t-product's forward and backward pass on a pre-built tensor, one CPU thread) takes 11–83 minutes; the exact evaluator needs under half a millisecond, 1.5 to 12 million times faster. These ratios depend on the reference t-product, an FFT implementation of the definition rather than an optimised kernel; the asymptotic gap between N^M and $\mathrm{nnz}(H_u)$ does not. Peak memory falls up to 213-fold, from 2.5 GiB to 12 MiB. Figure 2(a) plots the per-dataset values; Appendix M points to their run records. Beyond the reference hypergraphs, the exact evaluator reads House ($M = 81$) directly from its incidence matrix: after about 0.1 s of one-time preprocessing, a forward or backward pass takes under a millisecond on a GPU (Table 6). On synthetic hypergraphs with up to 51,200 nodes, inference time grows with the number of incidences times the width, as the cost bound of §3.3 predicts, and never with N^M (Table 7).

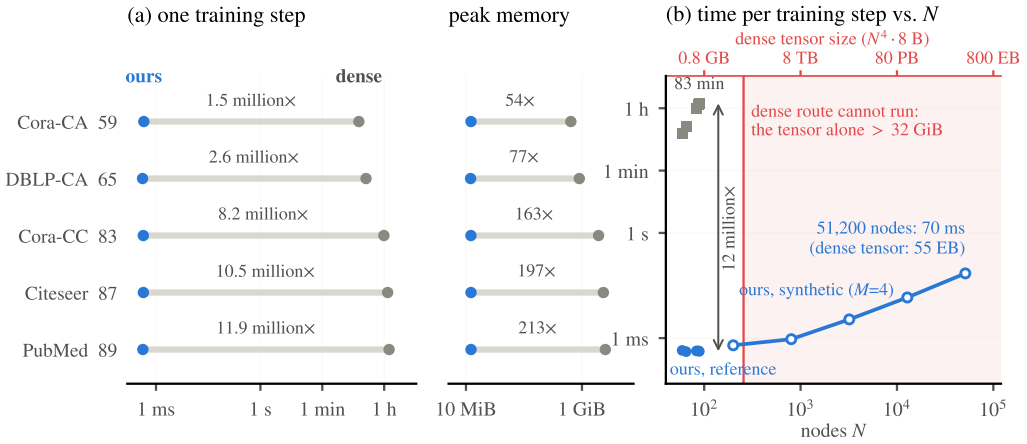


Figure 2: Exact evaluation is cheaper where the dense route runs and remains feasible beyond its reach. (a) On the five reference hypergraphs (§4; one CPU thread, $d = 64$), a training step takes 11–83 minutes on the dense route and under half a millisecond on the exact one, and peak memory drops by up to two orders of magnitude. (b) Time per training step against N at $M = 4$: beyond $N \approx 256$ the dense tensor alone exceeds 32 GiB, while the exact evaluator reaches 51,200 nodes (synthetic, width 32).

5 Limits of the sum readout

By Proposition 1, the released layer receives hypergraph structure only through $\bar{A}$. Weighted clique expansions are known to conflate distinct hypergraphs: Aksoy et al. (2024, §4.2.3) exhibit pairs that share one (graph trades), and Jiang et al. (2026) study the resulting loss of expressivity. The released layer inherits this limit exactly. With fixed features it has the form of the model in the next theorem (Remark 1): its encoder and the map g of (9), which may include a row-wise normalisation before W_1, act row by row, so relabelling only permutes the output rows, and S is unchanged (Appendix F).

Theorem 3 (Two-section indistinguishability). *Let $\widetilde{A}_H$ be $\bar{A}$ or $D^{\dagger}\bar{A}$, and consider the one-layer sum-readout model, with g any row-wise map,*

$$F_\theta(H, X_0) \;=\; g\Big(\widetilde{A}_H \, x \, \mathrm{diag}(S^{M-2})\Big), \qquad x = \mathrm{ReLU}(X_0 W_0 + \mathbf{1} b_0^\top), \quad S_c = \textstyle\sum_m x_{mc}. \tag{9}$$

If $\mathcal{H}_A, \mathcal{H}_B$ on N nodes share M and a permutation P gives

$$\widetilde{A}_{H_A} \;=\; P\,\widetilde{A}_{H_B}\,P^\top, \qquad X_A^{(0)} \;=\; P\,X_B^{(0)}, \tag{10}$$

then $F_\theta(H_A, X_A^{(0)}) = P\,F_\theta(H_B, X_B^{(0)})$ for every θ. For the normalised operator it suffices that $\bar{A}_A = P\bar{A}_B P^\top$ and

$$D_A \;=\; P\,D_B\,P^\top. \tag{11}$$

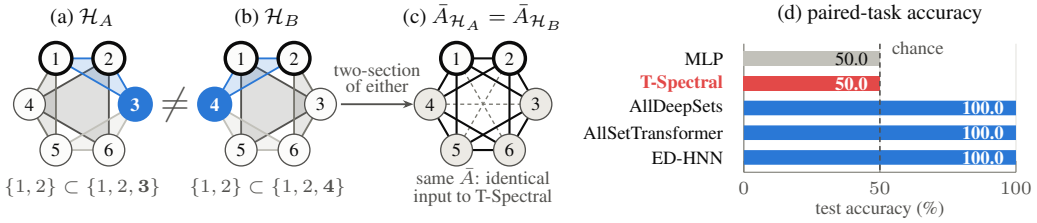


Figure 3: The labelled Pasch configuration. (a, b) Four triangles on six nodes; the anchor pair $\{1, 2\}$ is completed by node 3 in $\mathcal{H}_A$ and by node 4 in $\mathcal{H}_B$. (c) Both cover the same twelve pairs once, so $\bar{A}_{\mathcal{H}_A} = \bar{A}_{\mathcal{H}_B}$. (d) Accuracy on the query nodes: the released layer (T-Spectral in the panel) is held at chance (50%), while set-based models reach 100% (10 worlds × 3 initialisations; Table 9).

The proof is in Appendix F.1.

The Pasch witness. Figure 3 shows a labelled Pasch witness (Aryapoor, 2013): $\bar{A}_{\mathcal{H}_A} = \bar{A}_{\mathcal{H}_B}$, yet the pair $\{1, 2\}$ is completed by node 3 in $\mathcal{H}_A$ and by node 4 in $\mathcal{H}_B$. The two are isomorphic under the swap $(3\,4)$: one labelled trade, not two isomorphism classes, differing only in which labelled node completes the anchor pair (Appendix F.2). The features mark the anchor pair, and the label records whether a query node completes it. Models train on randomly relabelled copies of $\mathcal{H}_A$ and $\mathcal{H}_B$ and are tested on held-out copies, in 10 independently generated datasets, or *worlds* (Appendix F.3). Only the query nodes 3 and 4 are scored. In each hypergraph their $\bar{A}$ rows and features are identical but their labels opposite, so every model of the form (9) gives them the same output and is limited to 50%. T-Spectral (exact) and a clique-graph GCN both score exactly 50%; the GCN is our one-layer comparator (one propagation by the random-walk-normalised $0/1$ clique adjacency), not DHG-Bench's CEGCN. AllDeepSets, AllSetTransformer (Chien et al., 2022) and ED-HNN (Wang et al., 2023) score 100% (Table 9). On a control task whose label depends only on pairs, both $\bar{A}$-based models reach 100%, so the 50% is not a training failure (Appendix F.3).

Why the triple information cancels. The tensor records which triples exist. At $M = 3$ the summed convolution pairs every tube index a of A with every signal index l (before W): $Z_{ic} = \sum_j \sum_{a,l} A[i, j, a]\, x_{jc}\, x_{lc}$. The *matched* terms ($l = a$) sum to $Y_{ic} = \sum_j \sum_a A[i, j, a]\, x_{jc}\, x_{ac}$, the slice-matched T-Spatial layer, which weights each pair by the features of the node completing it and so sees triples; the *cross* terms ($l \neq a$) are $Z - Y$. With random features on the Pasch pair both parts differ between $\mathcal{H}_A$ and $\mathcal{H}_B$, but their sum $Z = S\,\bar{A}x$ does not: the triple-dependent parts cancel.

The printed (symmetrised) layer does not collapse. It reflection-pads each tube axis of the tensor and of the interaction signal before the t-product and removes the padding before the sum readout, so it is not a weighting of C (§3.4). Removing the padding turns each tube axis into a factor $S_c - \frac{1}{2}x_{a_r c}$, so for any tensor A the layer has the exact closed form (Appendix H)

$$Z^{\mathrm{sym}}[i, c] \;=\; \sum_{j,a} A[i, j, a]\, x_{jc} \prod_{r=1}^{T} \Big(S_c - \tfrac{1}{2} x_{a_r c}\Big) \;=\; \sum_{s=0}^{T} \Big(-\tfrac{1}{2}\Big)^{s} S_c^{T-s}\, B_s[i, c], \qquad (12)$$

where $B_s[i, c] = \sum_{j,a} A[i, j, a]\, x_{jc}\, e_s(x_{a_1 c}, \ldots, x_{a_T c})$ and e_s is the s-th elementary symmetric polynomial. The first block is $B_0 = \bar{A}x$, and the last is T-Spatial, $B_T = Y$. At $M = 3$, $Z^{\mathrm{sym}} = S_c\,\bar{A}x - \frac{1}{2}Y$: the matched terms, whose triple-dependent part cancels in the released layer, enter again with weight $-\frac{1}{2}$. (i) *Equivariance.* S_c is invariant under node relabelling and e_s is symmetric in its arguments, so every B_s, and hence Z^{sym}, is equivariant. (ii) *No collapse.* Every block beyond B_0 depends on which nodes complete each pair. On the Pasch pair, where the released layer's outputs coincide exactly, Z^{sym} differs between $\mathcal{H}_A$ and $\mathcal{H}_B$, and trained under the same protocol it scores 100% on the query nodes, though with small margins (Table 9; margins in Appendix F.3). The higher-order blocks are damped, however: for features of comparable magnitude each factor $x_{a_r c}/(2S_c)$ is $O(1/N)$, so B_s is suppressed by roughly N^{-s} relative to B_0. Evaluated over the tensor's nonzero tuples, (12) costs $O(T\,\mathrm{nnz}(A)\,d)$ time, and $\mathrm{nnz}(A)$ grows combinatorially with M (Appendix E.2).

6 Node classification at full scale

Theorem 3 bounds what the released layer can separate. We compare this one-hop model with hypergraph networks on the third-party DHG-Bench benchmark (Li et al., 2026), under its protocol: 10 seeds, a fresh random $50/25/25$ split per seed, and final-epoch test accuracy. T-Spectral (exact) runs matrix-free with every hyperedge at full size, on up to 88,860 nodes. Each baseline uses DHG-Bench's shipped configuration (generic defaults on Citeseer); ours is chosen on validation (Appendix J). DHG-Bench's House-committees differs from the archive's House.

The trained T-Spectral (exact) models propagate once, as the released layer does, and use the absorbed parameterisation of Remark 1, which keeps the hypothesis class of the layer on a fixed graph but not its training dynamics. On PubMed and the heterophilic datasets, the selected model also adds a row-wise normalisation before W_1; the released layer has no such operation (Appendix I).

Table 1: DHG-Bench node classification accuracy (%), final-epoch test, mean and 95% CI half-width over 10 seeds. T-Spectral (exact) ties the best baseline on the homophilic datasets and trails it on the heterophilic ones. The comparator is the test-best baseline, which favours the baselines. Configurations searched: one per baseline, DHG-Bench's shipped one ([†]none ships for Citeseer: generic defaults); 254–519 for ours, chosen on validation; 72 for TF-HNN and UniGCNII retuned on the homophilic sets (retuned rows: Table 14). OOM: out of GPU memory; —: not run. House-c./Walmart-t.: House-committees/Walmart-trips. Macro-F1: Table 13. [‡]Absorbed form of Remark 1 (W_1 free), plus a row-wise normalisation before W_1 on PubMed and the heterophilic datasets (Table 15); re-selected without it, those numbers change by -1.1 to $+0.2$ points (Appendix I).

	homophilic			heterophilic		
model	Cora	Citeseer[†]	PubMed	House-c.	Walmart-t.	Actor
MLP (features only)	$75.98_{\pm 0.84}$	$71.82_{\pm 0.88}$	$86.57_{\pm 0.23}$	$83.19_{\pm 0.88}$	$63.47_{\pm 0.20}$	$85.99_{\pm 0.41}$
HGNN	$78.49_{\pm 0.98}$	$70.09_{\pm 0.92}$	$86.14_{\pm 0.23}$	$78.79_{\pm 1.37}$	$77.49_{\pm 0.17}$	$77.87_{\pm 0.19}$
CEGCN	$77.27_{\pm 0.45}$	$66.28_{\pm 0.79}$	$86.03_{\pm 0.22}$	$57.21_{\pm 1.54}$	$70.46_{\pm 0.18}$	$67.48_{\pm 0.27}$
UniGCNII	$80.03_{\pm 0.78}$	$73.78_{\pm 0.70}$	$87.52_{\pm 0.20}$	$83.59_{\pm 0.84}$	$65.25_{\pm 0.53}$	$82.74_{\pm 0.39}$
AllSetTransformer	$78.35_{\pm 0.88}$	$66.82_{\pm 0.66}$	$87.69_{\pm 0.21}$	$83.19_{\pm 0.87}$	$78.74_{\pm 0.22}$	$85.87_{\pm 0.41}$
ED-HNN	$78.17_{\pm 0.85}$	$68.35_{\pm 0.61}$	$87.98_{\pm 0.24}$	$83.10_{\pm 0.85}$	OOM	$85.94_{\pm 0.27}$
TF-HNN	$80.03_{\pm 0.82}$	$72.55_{\pm 0.95}$	$88.03_{\pm 0.24}$	$82.60_{\pm 0.79}$	$77.33_{\pm 0.13}$	$85.60_{\pm 0.49}$
EHNN	$75.24_{\pm 1.17}$	$67.05_{\pm 0.76}$	$86.43_{\pm 0.61}$	$66.59_{\pm 4.93}$	—	$86.52_{\pm 0.25}$
T-Spectral (exact)[‡]	$80.40_{\pm 0.57}$	$74.41_{\pm 0.90}$	$88.53_{\pm 0.23}$	$79.81_{\pm 1.39}$	$77.10_{\pm 0.15}$	$82.79_{\pm 0.18}$

Homophilic datasets. On Cora, Citeseer and PubMed, where connected nodes tend to share a label, T-Spectral (exact) has the highest mean accuracy (Table 1), ahead of the best baseline by 0.37 to 0.63 points. We count these as ties, not wins: each margin is under one point and our arm had the larger search. Retuned with our selection rule over 72 configurations, TF-HNN and UniGCNII stay 0.18–0.58 points behind in accuracy, though both lead in macro-F1 on Citeseer (Appendix I).

Heterophilic datasets. On House-committees, Walmart-trips and Actor, where connected nodes often differ in label, T-Spectral (exact) trails the best baseline by 1.6 to 3.8 points. On House-committees and Actor, a feature-only MLP outperforms T-Spectral (exact) and most propagation models, so propagation helps little there.

7 Related work

HyperNATE (Bello et al., 2026) avoids the N^M cost with a new architecture; we keep the released layer fixed. Aksoy et al. (2024) evaluate hypergraph tensor contractions from incidence data; $\bar{A}$ is their TTSV2 at the all-ones vector (Appendix L). Theorem 3 places the released layer at the clique-expansion level (Kim et al., 2022; Zhang et al., 2025; Jiang et al., 2026).

Clique expansions date to Zhou et al. (2006) and Agarwal et al. (2006); most hypergraph networks propagate node→hyperedge→node (Feng et al., 2019; Yadati et al., 2019; Dong et al., 2020; Huang and Yang, 2021; Chien et al., 2022; Wang et al., 2023). Theorem 2 is an instance of equivariant-layer classification (Zaheer et al., 2017; Maron et al., 2019; Pearce-Crump, 2024), and the released layer resembles decoupled propagation (Wu et al., 2019; Gasteiger et al., 2019; Frasca et al., 2020).

8 Limitations and future work

Limitations. The benchmark results cover one layer and the unweighted sum readout only. Theorem 2 is proved at $M = 3$ only. Absorbing S^{M-2} into W_1 needs fixed input features, so the model-level statements and all trained results are transductive. The absorbed form keeps the hypothesis class of the layer with S^{M-2} on a fixed graph, not its training dynamics; training through that factor in the log domain (Appendix G.3) is untested. On DHG-Bench each baseline except the retuned TF-HNN and UniGCNII (homophilic sets only) runs one shipped configuration and ours the best of up to 519, favouring our arm, while the test-chosen best baseline favours theirs. As S and M

are global, disjoint hypergraphs batched together interact: one 81-node hyperedge moves the 3-node weights from $(0, \frac{1}{2})$ to about $(\frac{1}{3}, \frac{1}{3})$ (Table 8).

Future work. A *learned* mixture of the blocks of (12) could undo their N^{-s} damping.

9 Conclusion

Summing the circular convolution reduces the released layer to a one-hop clique-expansion model with a per-channel scale, evaluated exactly from incidence data on hypergraphs with 88,860 nodes or with hyperedges of 81 members. The same reduction exposes its loss of higher-order structure on the labelled Pasch task. The printed (symmetrised) layer retains higher-order information, with an evaluation cost that still depends on the tensor's nonzero tuples. The representation encodes the hypergraph; the readout determines what the layer can use.

Reproducibility Statement

The T-Spectral HyperGNN of Wang et al. (2025) is public (`https://github.com/wangfuli/T-HyperGNNs`); as it has no tagged release, we pin commit 0e3123b. Its datasets come from the archive linked in that repository's README, with SHA256 checksum

`f4a9435c29d3209a9dd1a753704eb9b19b3e0a685ef3348a0ddace9cf668f825`.

Seven of the nine distributed datasets are readable. Five of them can be built by the reference implementation; the other two (House, smallWalmart) are used only through our incidence route. The remaining two are truncated in the archive itself and were excluded rather than repaired (Appendix K). Every number in this paper is produced by a script that writes a run record keyed by its configuration; the supplementary `MANIFEST.tsv` maps each result and claim to its script and run records. The anonymised supplementary material contains the code, the run records, the exact-arithmetic nullspace certificates, a script that checks the proofs' worked examples, and one regeneration command per table and figure. The repository will be de-anonymised on acceptance.

Use of Large Language Models

Large language models were used in this work for literature search, reading the released source code, drafting and editing the manuscript, writing, adapting and running the experiment, control and verification scripts (including generating the labelled Pasch task), and critical feedback on mathematical statements, including drafting proof text. The authors led this work, reviewed the proofs and the code, and take full responsibility for every claim. The proofs of Proposition 1, Theorem 1 and Theorem 2 (at $M = 3$) were rewritten by an author from LLM drafts and then polished with an LLM; the proof of Theorem 3 was checked by an author. The Stirling-number code was written by an author for an earlier, unrelated project and adapted to this paper with an LLM. The exact enumeration at $M \in \{4, 5\}$ (Appendix D.2) is computer-assisted evidence, not a proof, and every machine-checked statement (parity, exactness, nullspace certificates) is backed by a script and a run record in the supplementary material.

References

Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 17–24, 2006. doi: 10.1145/1143844.1143847.

Sinan G. Aksoy, Ilya Amburg, and Stephen J. Young. Scalable tensor methods for nonuniform hypergraphs. *SIAM Journal on Mathematics of Data Science*, 6(2):481–503, 2024. doi: 10.1137/23M1584472. arXiv:2306.17825.

Masood Aryapoor. The Pasch configuration and Steiner triple systems, 2013. arXiv:1306.1257.

Anirban Banerjee, Arnab Char, and Bibhash Mondal. Spectra of general hypergraphs. *Linear Algebra and its Applications*, 518:14–30, 2017. doi: 10.1016/j.laa.2016.12.022.

Nicolás Bello, Fuli Wang, Daniel L. Lau, and Gonzalo R. Arce. HyperNATE: Scaling tensor-based hypergraph neural networks through attention. *Neural Networks*, 203:109139, 2026. doi: 10.1016/j.neunet.2026.109139. Article available ahead of the assigned November 2026 issue; a precursor appears as SSRN preprint, DOI 10.2139/ssrn.5225175.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Yihe Dong, Will Sawin, and Yoshua Bengio. HNHN: Hypergraph networks with hyperedge neurons, 2020. arXiv:2006.12278; ICML Graph Representation Learning and Beyond workshop.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558.

Fabrizio Frasca, Emanuele Rossi, Davide Eynard, Benjamin Chamberlain, Michael Bronstein, and Federico Monti. SIGN: Scalable inception graph neural networks, 2020. arXiv:2004.11198; ICML Graph Representation Learning and Beyond workshop.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized PageRank. In *International Conference on Learning Representations (ICLR)*, 2019.

Jing Huang and Jie Yang. UniGNN: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2563–2569, 2021. doi: 10.24963/ijcai.2021/353.

Fengqing Jiang, Yuetai Li, Yichen Feng, Kaiyuan Zheng, Luyao Niu, Bhaskar Ramasubramanian, Basel Alomair, Linda Bushnell, and Radha Poovendran. The WidthWall: A strict expressivity hierarchy for hypergraph neural networks, 2026. arXiv:2605.13690v1, preprint.

Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011. doi: 10.1016/j.laa.2010.09.020.

Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013. doi: 10.1137/110837711.

Jinwoo Kim, Saeyoon Oh, Sungjun Cho, and Seunghoon Hong. Equivariant hypergraph neural networks. In *European Conference on Computer Vision (ECCV)*, 2022. arXiv:2208.10428; code github.com/jw9730/ehnn.

Fan Li, Xiaoyang Wang, Wenjie Zhang, Ying Zhang, and Xuemin Lin. DHG-Bench: A comprehensive benchmark for deep hypergraph learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Edward Pearce-Crump. Connecting permutation equivariant neural networks and partition diagrams. In *27th European Conference on Artificial Intelligence (ECAI)*, pages 1511–1518, 2024. doi: 10.3233/FAIA240655.

Karelia Peña-Peña, Daniel L. Lau, and Gonzalo R. Arce. t-HGSP: Hypergraph signal processing using t-product tensor decompositions. *IEEE Transactions on Signal and Information Processing over Networks*, 9:329–345, 2023. doi: 10.1109/TSIPN.2023.3276687.

Bohan Tang, Zexi Liu, Keyue Jiang, Siheng Chen, and Xiaowen Dong. Training-free message passing for learning on hypergraphs. In *International Conference on Learning Representations (ICLR)*, 2025.

Fuli Wang, Karelia Peña-Peña, Wei Qian, and Gonzalo R. Arce. T-HyperGNNs: Hypergraph neural networks via tensor representations. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5044–5058, 2025. doi: 10.1109/TNNLS.2024.3371382.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. Equivariant hypergraph diffusion neural operators. In *International Conference on Learning Representations (ICLR)*, 2023.

Felix Wu, Amauri H. Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 6861–6871, 2019.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 3391–3401, 2017.

Detian Zhang, Chengqiang Zhang, Yanghui Rao, Li Qing, and Chunjiang Zhu. Improved expressivity of hypergraph neural networks through high-dimensional generalized Weisfeiler-Leman algorithms. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 76880–76908, 2025.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19 (NeurIPS)*, pages 1601–1608, 2006.

A NOTATION

Table 2: Notation used throughout. Two letters are overloaded by long-standing convention and are always disambiguated by their arguments: c is a hyperedge size in $a_{M,c}$, $b_{M,c}$, $L_{M,c}$ and a feature channel in x_{jc}, S_c; $S(\cdot,\cdot)$ is a Stirling number and S (or S_c) a channel sum; a is a tube position in (3) and the vector of self weights $a_{M,|e|}$ in (7); b is the vector of pair weights and b_0 the encoder bias; d is the feature width.

symbol	meaning
V, N	node set $\{1,\dots,N\}$ and its size
E, e, $c=\lvert e\rvert$	hyperedges (a multiset of node-sets), one hyperedge, its size
$M=\max_e \lvert e\rvert$, $T=M-2$	order of the hypergraph; number of tube axes
H, H_u, $\mathcal{E}_u$, h_e	full and deduplicated incidence matrices; distinct node-sets; a column of H_u
$\mathrm{nnz}(\cdot)$	number of nonzero entries
$D=\mathrm{diag}(H\mathbf{1})$, $D^\dagger$	degree matrix (with multiplicity); its pseudo-inverse
A, $A^{[k]}$	order-M adjacency tensor; its $N\times N$ slice at tube position $k\in[N]^T$
A_0, A^{norm}	unnormalised tensor A; its normalised form $D^\dagger A_0$ (also A_{norm})
$L_{M,c}=c!\,S(M,c)$	number of length-M tuples covering a size-c node-set
X_0, W_0, b_0	input features and encoder parameters
$x\in\mathbb{R}^{N\times d}$, d	encoded features $\mathrm{ReLU}(X_0W_0+\mathbf{1}b_0^\top)$; feature width
$X^{[k]}$, $C^{[k]}$	interaction-signal slice and t-product output slice at tube position k
$S=x^\top\mathbf{1}$, S_c	channel sums
W	channel mixer applied after the readout
Z, Z_w, Y	sum readout; weighted readout with tube weights w; T-Spatial output
w, T_w	tube weights of a readout (Definition 1); their Hankel matrix (Lemma 2)
U	propagated features before the scale S^{M-2} (algorithm of §3.3)
$\bar{A}$, $\bar{A}^{\mathrm{norm}}$	tube-summed adjacency $\sum_a A[\cdot,\cdot,a]$; its degree-normalised form $D^\dagger\bar{A}$
$a_{M,c}$, $b_{M,c}$	self and pair weights of a size-c hyperedge at order M (Theorem 1)
B_s	degree-s block of the printed (symmetrised) layer (Appendix H)
P	permutation matrix

B PROOF OF THEOREM 2 AT $M = 3$

Setting. At $M = 3$ there is one tube axis, indexed by $\{0, \dots, N-1\}$ modulo N. Fix one channel (W commutes with the readout). An index-keyed weighting (Definition 1) is a fixed $w \in \mathbb{R}^N$, the readout is $Z_w[i] = \sum_k w_k\, C[i,k]$, and the released readout is $w = \mathbf{1}$. The proof is for the unnormalised builder A_0; Remark 2 covers the normalised one. The qualifier "up to $Z_w \equiv 0$" is vacuous at $M = 3$: the argument of Proposition 2 with T_w in place of D shows that $Z_w \equiv 0$ forces $w = 0$.

The idea. The readout feeds the features through a fixed $N \times N$ matrix T_w. Equivariance forces T_w to be unchanged by every relabelling, so $T_w = \alpha I + \beta J$ (Lemma 1); but T_w is a Hankel matrix, and the only Hankel matrices of that form are constant. In Fourier terms, a non-constant w reads some nonzero frequency, which weights tube position k by a phase tied to the label k; only the zero frequency, the collapse, is label-free.

Lemma 1 (Commutant)**.** *Let X be a square matrix indexed by a finite set V, $|V| \geq 2$. If $X_{\pi i, \pi j} = X_{ij}$ for every permutation π of V (equivalently $XP = PX$ for every permutation matrix P), then $X = \alpha I + \beta J$, J the all-ones matrix.*

Proof. A permutation moves any diagonal position to any other, and any off-diagonal position to any other. □

The lemma is used twice: here, and for Theorem 1 (Appendix C), where each node-set's block of $\bar{A}$ is invariant under permutations of that node-set.

Step 0: the readout as a matrix. By (3), $C[i,k] = \sum_{j,a} A[i,j,a]\, x_j\, x_{(k-a) \bmod N}$, so

$$Z_w[i] = \sum_{j,a} A[i,j,a]\, x_j\, (T_w x)[a], \qquad (T_w x)[a] = \sum_k w_k\, x_{(k-a) \bmod N}. \tag{13}$$

Lemma 2 (Hankel form)**.** $(T_w)_{a,m} = w_{(a+m) \bmod N}$.

Proof. Substitute $m = (k - a) \bmod N$. □

At $N = 3$, $T_w = \begin{pmatrix} w_0 & w_1 & w_2 \\ w_1 & w_2 & w_0 \\ w_2 & w_0 & w_1 \end{pmatrix}$. For $w = \beta\mathbf{1}$, $T_w = \beta J$ and $(T_w x)[a] = \beta S$: Proposition 1.

Proposition 2. *Let $N \geq 5$. Z_w is equivariant on every builder input of order 3 if and only if $T_w P = P T_w$ for every permutation matrix P.*

Proof. Let π have matrix P, $(Pv)_{\pi(i)} = v_i$; the relabelled tensor is $A^\pi[\pi i, \pi j, \pi a] = A[i,j,a]$. Substituting $j' = \pi j$, $a' = \pi b$ in (13) gives $Z_w(A^\pi, Px)[\pi i] = \sum_{j,b} A[i,j,b]\, x_j\, (P^{-1} T_w P x)[b]$, so with $D := P^{-1} T_w P - T_w$ equivariance reads

$$\textstyle\sum_{j,b} A[i,j,b]\, x_j\, (Dx)[b] = 0 \quad \text{for every head } i, \text{ feature } x \text{ and builder input.}$$

If $D = 0$ this holds. Conversely, take nodes $u \neq v$, the input $\{\{u,v\}, g\}$ with g a 3-node edge avoiding u, v (it fixes the order at 3; this needs $N \geq 5$), and $x = e_v$. Only tuples with middle entry v survive, none from g: the pair edge writes $\frac{1}{3}$ into (v,v,u), (u,v,u), (u,v,v). Head v gives $\frac{1}{3} D_{uv} = 0$ and head u gives $\frac{1}{3}(D_{uv} + D_{vv}) = 0$. As u, v are arbitrary, $D = 0$. □

Proposition 3. *For $N \geq 3$, if $T_w = \alpha I + \beta J$ then $w \equiv \beta$.*

Proof. By Lemma 2 every w_s sits at an off-diagonal position of T_w: $(0, s)$ if $s \neq 0$, and $(1, N-1)$ if $s = 0$. All off-diagonal entries equal β. □

Conclusion. Equivariance $\Leftrightarrow$ T_w commutes with every P (Proposition 2) $\Rightarrow$ $T_w = \alpha I + \beta J$ (Lemma 1) $\Rightarrow$ $w \equiv \beta$ (Proposition 3). Conversely $w \equiv \beta$ gives $T_w = \beta J$ and $Z_w = \beta\, \bar{A}\, x\, S$, which is (4) at $M = 3$. This proves Theorem 2. □

Remark 2 (The source-normalised builder)**.** The proof applies verbatim to $A_{\mathrm{norm}} = D^{\dagger} A_0$ (Appendix D.1), which is also relabelling-covariant: in the test input u, v have degree 1 and the three tuples start at u or v, so their entries are unchanged. For constant w the output is $\beta\, D^{\dagger} \bar{A}_0\, x\, S$. The worked-example checker of the supplementary material (Appendix M) also checks both builders numerically.

The hypothesis $N \geq 5$, and the enumeration domain. The test input needs $N \geq 2 + 3 = 5$ nodes, so Table 3's $M = 3$, $N \in \{3, 4\}$ rows are exact enumeration, not proof. That enumeration, like the $M \in \{4, 5\}$ rows, runs over single-edge tensors $A_e^{(M)}[t] = \frac{c}{L_{M,c}} \mathbf{1}[t \in \Pi(e)]$; for $N \geq M + c$ these are differences of builder inputs, $A^{(M)}_{\{g,e\}} - A^{(M)}_{\{g\}}$ with g an M-edge disjoint from e, and the constraints are linear in A, so the domain adds no constraint that builder inputs do not.

The convolution pairing. Lemma 2 is for the correlation kernel $w_{(a+m) \bmod N}$ that $\sum_k w_k C[\cdot, \cdot, k]$ forms; pairing by convolution replaces w by its reversal, an invertible change of parameters that maps constants to constants, so it gives the same answer. The difference-index kernel $w_{(a-m) \bmod N}$ is a different operator and not a readout of the released layer (exact enumeration gives dimension $M - 1$; certificates in the supplementary material).

C Proof of Theorem 1

The idea. Read one node-set's block of $\bar{A}$ as a probability table: $\bar{A}^{(e)}[i, j]$ is $|e|$ times the chance that a uniformly random surjective tuple on e starts with (i, j). The table cannot tell the nodes of e apart, so it has one diagonal value a and one off-diagonal value b (Lemma 1); its rows sum to 1, and a is the chance that the first two slots agree.

Setup and conventions. $\mathcal{H} = (V, E)$ with $N = |V|$ and E a finite *multiset* of node-sets; $M = \max_{e \in E} |e|$ (the reference `HyperRepresentation.__init__` sets $M = \max_e \mathbf{1}^{\top} h_e$) and $T = M - 2 \geq 0$. For $\tau \in [N]^M$, $\mathrm{supp}(\tau)$ is its set of distinct entries. $S(n, k)$ is the Stirling number of the second kind; $k!\, S(n, k)$ counts the surjections $[n] \twoheadrightarrow [k]$. The builder writes, for each e with $c = |e|$ and each τ with $\mathrm{supp}(\tau) = e$, the value $A[\tau] \leftarrow c / L_{M,c}$ *by assignment*, where $L_{M,c} = |\{\tau \in e^M : \mathrm{supp}(\tau) = e\}|$; $\bar{A}[i, j] = \sum_{a \in [N]^T} A[i, j, a]$. $\mathcal{E}_u = \{\mathrm{supp}(e) : e \in E\}$ is the set of distinct node-sets, $H_u \in \{0, 1\}^{N \times |\mathcal{E}_u|}$ its incidence matrix with columns h_e, and $H \in \{0, 1\}^{N \times |E|}$ the full (non-deduplicated) incidence matrix.

Singleton hyperedges. The reference builder admits a size-1 column ($c = 1$) and pads it by repetition to the single tuple $(i, \ldots, i)$, so we treat $c = 1$ explicitly ("Singletons" below). Genuine hyperedges have $2 \leq c \leq M$: $c > M$ cannot occur by the definition of M, and $c \geq 2$ makes $b_{M,c}$ non-vacuous.

Isolated nodes. $D^{\dagger}$ is the Moore–Penrose pseudo-inverse of the diagonal D ($d_i^{\dagger} = 0$ if $d_i = 0$). An isolated node has a zero row in H_u, hence in $\bar{A}$, so $D^{\dagger} \bar{A}$ leaves that row zero and no $0/0$ is formed.

The builder's tuple set. For e with $|e| = c$, `generate_perms`(e) returns exactly $\Pi(e) = \{\tau \in e^M : \mathrm{supp}(\tau) = e\}$, each tuple once. If $c = M$ it returns the $M!$ orderings of e, which are exactly the $\tau \in e^M$ with distinct entries. If $c < M$, for every size-$(M - c)$ multiset m drawn with replacement from e it returns `set(permutations(`$m \uplus e$`))`; each such τ lies in e^M and uses every node of e. Conversely, deleting one copy of each node of e from the entries of a $\tau \in \Pi(e)$ leaves a unique $m(\tau)$, so τ is produced in branch $m(\tau)$ only, and `set` de-dupes within a branch. So $|\Pi(e)| = L_{M,c}$ counts the surjections $[M] \twoheadrightarrow e$: split the M slots into c unnamed nonempty groups ($S(M, c)$ ways) and name the groups by the nodes of e ($c!$ ways), which is (2); the alternating sum there is inclusion–exclusion over the nodes left out.

Proof. Fix $e \in \mathcal{E}_u$ with $c = |e| \geq 2$, and let τ be uniformly random in $\Pi(e)$.

1. *One block per node-set.* $A[\tau] = \sum_{e \in \mathcal{E}_u} \frac{|e|}{L_{M,|e|}} \mathbf{1}[\tau \in \Pi(e)]$, so $\bar{A} = \sum_{e \in \mathcal{E}_u} \bar{A}^{(e)}$.
   Why: a tuple determines its own support, so it lies in $\Pi(e)$ for exactly one node-set. Different node-sets never write a common cell, and a repeated node-set rewrites the identical value (assignment), so multiplicities in E are invisible.

2. *A probability table.* $\bar{A}^{(e)}[i,j] = \frac{c}{L_{M,c}} \#\{\tau \in \Pi(e) : \tau_1 = i, \tau_2 = j\} = c \cdot \Pr[\tau_1 = i,\ \tau_2 = j]$, which vanishes unless $i, j \in e$.
   Why: the tube sum counts the tuples that start with (i, j).

3. *Two values.* $\bar{A}^{(e)} = b\, h_e h_e^\top + (a - b)\, \mathrm{diag}(h_e)$ for some a, b.
   Why: renaming the nodes of e maps $\Pi(e)$ onto itself, so the table on $e \times e$ is invariant under every permutation of e; apply Lemma 1.

4. *Rows sum to one.* $\sum_j \bar{A}^{(e)}[i,j] = c \Pr[\tau_1 = i] = 1$ for $i \in e$, so $a + (c-1)b = 1$.
   Why: by the same symmetry each node of e is equally likely to come first.

5. *The diagonal.* $a = \Pr[\tau_1 = \tau_2] = L_{M-1,c}/L_{M,c} = S(M-1,c)/S(M,c) = a_{M,c}$, and then $b = (1-a)/(c-1) = b_{M,c}$.
   Why: the trace is $c\,a$ and also $c \Pr[\tau_1 = \tau_2]$; merging the first two slots maps $\{\tau \in \Pi(e) : \tau_1 = \tau_2\}$ bijectively onto the surjections of $M - 1$ slots onto e.

Summing the blocks gives (6), and (7) follows from $\sum_e b_{M,|e|} h_e h_e^\top = H_u \mathrm{diag}(b) H_u^\top$ and $\sum_e (a_{M,|e|} - b_{M,|e|}) h_e = H_u(a - b)$.

Singletons. If $|e| = 1$, the only tuple is $(i, \ldots, i)$, with value 1, so $\bar{A}^{(e)} = \mathrm{diag}(h_e)$: $a_{M,1} = 1$, $b_{M,1} = 0$. *Graphs.* At $M = 2$, $a_{2,2} = 0$ and $b_{2,2} = 1$ (Corollary 2).

Source normalisation. `Adjacency_normalized` divides cell τ by $d_{\tau_1} = \sum_k H_{\tau_1 k}$ — a function of the *first* coordinate only, read off the *full* incidence matrix (line `d_i1 = sum(self.H[i1,:])`, no deduplication). The tube-sum fixes $\tau_1 = i$, so d_i is constant across the sum and factors out as a left row-scaling:

$$\bar{A}^{\mathrm{norm}} = D^\dagger \bar{A}, \qquad D = \mathrm{diag}(H\mathbf{1}), \quad (H\mathbf{1})_i = \#\{e \in E : i \in e\} \text{ (with multiplicity)},$$

with $D^\dagger$ the pseudo-inverse convention fixed above. The numerator depends only on $\mathcal{E}_u$ (item 1), the denominator on the full multiset E; both are required for bit-exact parity with the reference. The left factor is not symmetric, so $\bar{A}^{\mathrm{norm}}$ is not head-symmetric — the $h_e h_e^\top$ form is read on $\bar{A}_0 = \bar{A}$. ■

Sign of the diagonal correction. For every $2 \le c \le M$, $a_{M,c} < \frac{1}{c} < b_{M,c}$: a surjective tuple repeats its first entry less often than a uniform one. Classifying the surjections of M slots by the last slot (either the first $M - 1$ slots already cover e, or they miss exactly the last slot's node) gives $L_{M,c} = c\,(L_{M-1,c} + L_{M-1,c-1}) > c\,L_{M-1,c}$, so $a_{M,c} < \frac{1}{c}$ and $b_{M,c} = \frac{1 - a_{M,c}}{c-1} > \frac{1}{c}$. So $\mathrm{diag}(H_u(a-b))$ is strictly negative at every node in a node-set of size at least 2, while $\bar{A}$ is entrywise nonnegative.

Cost of evaluating with $\bar{A}$. Applied to $X \in \mathbb{R}^{N \times d}$, $\bar{A}X = H_u \mathrm{diag}(b)(H_u^\top X) + (H_u(a-b)) \odot X$: three sparse products, $O((\mathrm{nnz}(H_u) + N)\, d)$ time and $O(\mathrm{nnz}(H_u) + (N + |\mathcal{E}_u|)d)$ memory, forming no order-M tensor and no clique-expansion matrix. The one-time build hashes the sorted support of each incidence column, $O(\mathrm{nnz}(H))$ expected time, and costs one Stirling recurrence per hyperedge size that occurs.

Overflow-safe coefficients. The integers in (2)–(5) grow super-exponentially: at $M = 81$ and $c = 40$, $L_{M,c}$ has 127 decimal digits. We compute $S(M,c)$ by the integer recurrence $S(n,k) = k\,S(n-1,k) + S(n-1,k-1)$, form $a_{M,c}$ and $b_{M,c}$ as exact rationals, and round to double precision once, which keeps (14) exact up to that single rounding. Tests pin the exact values at $M \in \{4, 5, 81\}$ against brute-force tuple enumeration, an independent symbolic evaluation, and the closed forms $a_{M,2} = \frac{2^{M-2}-1}{2^{M-1}-1}$, $a_{M,M} = 0$, $b_{M,M} = \frac{1}{M-1}$.

Table 3: Exact commutant nullspace over the admissible domain, with exact rational certificates in the supplementary material. "eq. dim" is the equivariant dimension after quotienting the zero-layer subspace $\ker L$ (see text), under the correlation pairing. Rows with $N < M$ are excluded, since the admissible family does not determine w there. "proof" marks the rows inside the hypotheses of Theorem 2 ($M = 3$, $N \geq 5$); all other rows are exact enumeration.

M	N	tube dim N^{M-2}	zero-layer subspace	eq. dim (corr.)	evidence
3	3	3	0	1	exact enumeration
3	4	4	0	1	exact enumeration
3	5	5	0	1	proof (App. B)
3	6	6	0	1	proof (App. B)
4	4	16	6	1	exact enumeration
4	5	25	10	1	exact enumeration
4	6	36	15	1	exact enumeration
5	5	125	90	1	exact enumeration
5	6	216	160	1	exact enumeration

D Equivariance and the absorbed parameterisation

D.1 Mode symmetry and source normalisation

The unnormalised tensor A_0 is symmetric in all M modes jointly. The source normalisation of the reference implementation (§2) forms $A_{\mathrm{norm}} = D^{\dagger} A_0$ with $D = \mathrm{diag}(H\mathbf{1})$. The left factor touches only the first (head) mode, so A_{norm} keeps joint symmetry across the other $M - 1$ modes, the second head index and the T tube axes, but not across the first. Hence $\bar{A}_0$ is symmetric while $\bar{A}^{\mathrm{norm}} = D^{\dagger} \bar{A}_0$ is not, and the symmetric $h_e h_e^{\top}$ form of Theorem 1 is stated for A_0. The other results are unaffected by the left factor: the collapse (Proposition 1) needs no tensor symmetry, and Theorem 3 is stated for both operators. Theorem 2 is stated for A_0, and its $M = 3$ proof also covers A_{norm} (Remark 2). The enumeration of Table 3 runs over the unnormalised single-edge domain, so at $M \geq 4$ it does not cover the normalised builder.

D.2 Enumeration beyond $M = 3$ and the zero-layer quotient

Computational Observation 1 (exact enumeration at $M \in \{4, 5\}$). For the pairs $(M, N) \in \{(4,4), (4,5), (4,6), (5,5), (5,6)\}$, exact enumeration of the commutant nullspace over the admissible domain returns, modulo the zero-layer subspace, an equivariant space of dimension exactly 1 under the correlation pairing (Table 3; exact rational certificates in the supplementary material). This is computer-assisted evidence from exact enumeration, not a written proof, and it covers only those five pairs.

Conjecture 1 (extension to higher order). For every $M \geq 3$ and $N \geq M$, the correlation-pairing conclusion of Theorem 2 holds: modulo the zero-layer subspace the equivariant weightings form a space of dimension exactly 1. The proof of Appendix B does not extend axis by axis. A relabelling permutes all T tube axes at once, and the commutant of this diagonal action, indexed by set partitions of the axes (Maron et al., 2019), is strictly larger than the tensor product of the single-axis commutants of Lemma 1. Moreover, the builder's tensor is symmetric in all M indices jointly, so no admissible input varies one tube axis while holding the others fixed.

The zero-layer quotient. On the admissible domain (the unnormalised single-edge tensors of Appendix B), a weighting in $\ker L = \{w : Z_w \equiv 0\}$ is equivariant only because it computes nothing, so the classification is stated modulo $\ker L$. Because A is symmetric in its tube axes, every w whose tube factor has vanishing symmetric part lies in $\ker L$. These directions number $N^{M-2} - \binom{N+M-3}{M-2}$, a lower bound on $\dim \ker L$. At $M \geq 4$ a nonconstant w can therefore compute exactly the constant readout; at $M = 4$, $w_{ab} = 1 + u_a v_b - v_a u_b$ does so for any $u, v \in \mathbb{R}^N$, so the precise claim is $[w] \neq [\beta \mathbf{1}]$ in the quotient. That the symmetry-null directions exhaust $\ker L$, so that the quotient has dimension one, is verified for the nine pairs of Table 3 by an independent checker that rebuilds both kernels in exact rational arithmetic; beyond them it is Conjecture 1.

D.3 THE ABSORBED PARAMETERISATION

Remark 3 (What the released model computes)**.** Put the encoder of §2 in front. By (4) with $w \equiv 1$ and $x = \mathrm{ReLU}(X_0 W_0 + \mathbf{1} b_0^\top)$, and since W commutes through the tube sum (§3.1), the model output is $ZW = \bar{A}\, x\, \mathrm{diag}(S)^{M-2}\, W$ with $S_c = \sum_m x_{mc}$. In the transductive setting X_0 is fixed, so at fixed encoder parameters $\mathrm{diag}(S)^{M-2}W$ is a single matrix W_1 and the model realises $X_0 \mapsto \bar{A}\,\mathrm{ReLU}(X_0 W_0 + \mathbf{1} b_0^\top)\, W_1$. Conversely every W_1 is reachable: where $S_c > 0$ the diagonal is invertible on that channel, and where $S_c = 0$ the channel $x_{\cdot c}$, a sum of ReLU outputs equal to zero, is identically zero, so the corresponding row of W_1 acts on nothing. The released model is therefore *exactly* a one-hidden-layer ReLU encoder, one linear propagation by the fixed matrix $\bar{A}$, and a linear map (pre-softmax logits; dropout and $\log \texttt{softmax}$ follow): the order-M tensor contributes $\bar{A}$ and a per-channel scalar, and nothing else. The same holds for $D^\dagger \bar{A}$, since $D^\dagger$ does not involve S. The converse needs nonnegative channels and fixed X_0: for signed x, $S_c = 0$ need not imply $x_{\cdot c} = 0$.

E COST AND COEFFICIENTS OF THE EXACT EVALUATOR

E.1 ROW MASS

Corollary 1 (Row mass)**.** *For every* $2 \le c \le M$*,*

$$a_{M,c} + (c-1)\, b_{M,c} = 1, \tag{14}$$

so row i *of* $\bar{A}$ *sums to the number of distinct node-sets containing* i*, and* $\bar{A}^{\mathrm{norm}}$ *is row-stochastic whenever no node-set is repeated (proof in Appendix E).*

This is the degree property Wang et al. (2025) designed their tensor to have.

Proof. Item 4 of Appendix C is (14): each block $\bar{A}^{(e)}$ has rows summing to 1, with $a_{M,c}$ on the diagonal and $b_{M,c}$ at the $c-1$ other positions of e (a singleton adds its unit self-loop). Summing the blocks, row i of $\bar{A}$ sums to $\#\{e \in \mathcal{E}_u : i \in e\}$. If no node-set repeats, this is $d_i = (H\mathbf{1})_i$, so $D^\dagger \bar{A}$ has unit row sums at every non-isolated node (isolated rows are zero). In the running example the row sums are $(2, 2, 1)$. □

Corollary 2 (Graphs)**.** *At* $M = 2$*,* $a_{2,2} = 0$ *and* $b_{2,2} = 1$*: for a graph,* $\bar{A}$ *is its* $0/1$ *adjacency matrix* A *and* $\bar{A}^{\mathrm{norm}} = D^{-1}A$ *the random-walk matrix.*

E.2 COST MODEL OF ONE LAYER

Table 4 compares the two routes of §3.3. Storing only the nonzero tuples does not rescue the dense route: `House` has $\approx 1.25 \times 10^{134}$ of them.

E.3 COST AS A FUNCTION OF THE ORDER M

M is set by the data through (1), so the N^M tensor is not a tunable expense. Table 5 gives, for every hypergraph distributed with the reference implementation, the order and the array its construction allocates.

Table 4: Cost of one T-Spectral layer. Dense figures as stated by Wang et al. (2025); $\mathrm{nnz}(H_u)$ counts incidences after deduplication; `House` values from Table 5 and this appendix. Measured times are in §4.

	dense tensor route	incidence route (ours)
stored structure	N^M tensor entries	$\mathrm{nnz}(H_u)$ incidences, $2\lvert\mathcal{E}_u\rvert$ coefficients
one-time build	all $\sum_e L_{M,\lvert e\rvert}$ padded tuples	deduplication, $O(\mathrm{nnz}(H))$ expected
propagation	$O(dN^M)$, plus FFTs	$O((\mathrm{nnz}(H_u) + N)\, d)$
tube intermediates	$N^{M-1}d$ interaction signal	d channel sums S
`House` ($N = 1290$, $M = 81$)	$\sim 7.26 \times 10^{252}$ B	11,809 incidences

Table 5: Order census of the nine distributed hypergraphs. M is $\max_e |e|$ by (1), a property of the data alone, independent of which implementation processes it; the allocation column is a dense `float64` array of N^M entries. "builds" is whether the reference construction completes at all. `House`'s $M = 81$ exceeds what the released signal einsum supports ($M \leq 5$); reaching it uses our exact-rational coefficient port (whose output is Table 8), not released code. Two datasets could not be read (Appendix K).

dataset	N	E	$\mathrm{nnz}(H)$	M	N^M array	builds
`coauthorship/cora`	59	40	106	4	92.45 MiB	yes
`coauthorship/dblp`	65	29	88	4	136.19 MiB	yes
`cocitation/cora`	83	42	125	4	362.08 MiB	yes
`cocitation/citeseer`	87	50	135	4	437.09 MiB	yes
`cocitation/pubmed`	89	40	122	4	478.69 MiB	yes
`new/smallWalmart`	18032	5798	28990	5	$\sim 1.53 \times 10^{22}$ B	no
`new/House`	1290	341	11841	81	$\sim 7.26 \times 10^{252}$ B	no
`3dObject/ModelNet40`, `3dObject/NTU` — not readable, see Appendix K						

Table 6: Matrix-free incidence evaluator on the full-scale archive datasets (M cross-checked against the order census). One-layer model, d=64, 3 seeds, RTX 3060 Ti; forward/backward are the per-seed median. $\mathrm{nnz}(H_u)$ counts incidences after deduplication (Table 5 lists the pre-deduplication $\mathrm{nnz}(H)$: 11,841 and 28,990).

dataset	M	$\mathrm{nnz}(H_u)$	preproc. (ms)	fwd (ms)	bwd (ms)	GPU peak (MiB)
`House`	81	11,809	108.14	0.220	0.410	26.58
`smallWalmart`	5	28,960	160.64	0.432	1.003	59.50

The construction completes on five of the nine hypergraphs, all with $N \leq 89$ and $M = 4$, one to two orders of magnitude smaller than the standard versions of the same data, which Wang et al. (2025) subsampled so that the $O(N^M)$ models would fit. Order three, the reference repository's default, occurs on none of them, which is why the enumeration at $M \in \{4, 5\}$ matters (Appendix D.2). The incidence route needs only the 336 distinct node-sets of `House` ($\mathrm{nnz}(H_u) = 11{,}809$; the $\mathrm{nnz}(H) = 11{,}841$ of Table 5 counts all 341 hyperedges).

E.4 Measured cost of the exact evaluator

Table 6 times the matrix-free evaluator on the two full-size archive hypergraphs; Table 7, which backs Figure 2(b), times it on synthetic hypergraphs with 4-node hyperedges, hidden width 32, in single precision. Preprocessing is a one-time cold cost, timed apart from the warm passes. In the sweep, forward and backward are medians of 500 repetitions, an epoch of 50, and end-to-end training of 5 calls of 20 epochs, one subprocess per row. The *materialised* arm applies an explicit sparse $\bar{A}$, the *incidence* arm is matrix-free. GPU "coef" includes one-time CUDA initialisation; the CPU resident-set change was below page granularity ("n/m").

GPU forward time grows about eightfold, from 0.18 to 1.47 ms, while $\mathrm{nnz}(H_u)$ grows sixteenfold from 25,600: no faster than $\mathrm{nnz}(H_u)\, d$, and nowhere near N^M; on CPU it grows somewhat faster than $\mathrm{nnz}(H_u)$. The largest run peaks at 201 MiB. With 4-node hyperedges the incidence arm uses slightly more GPU memory than the materialised arm, and the cold build (10 ms to 2.8 s) is comparable to or larger than a 20-epoch training block.

E.5 Duplicate node-sets and assignment semantics

The reference builder writes `A[perm] = |e| / #perms` by *assignment*. Distinct node-sets never write a common tuple (item 1 of Appendix C), so a tuple is written twice only when a node-set occurs more than once in E, and the second write stores the same value again. Accumulating instead of assigning would count such a node-set once per copy. On `coauthorship/cora`, one node-set is repeated, so 14 of 948 distinct tuples are written twice, and $\bar{A}$ under the two semantics differs by 35% relative in max norm; the other four datasets have no repeated node-set and the two

Table 7: Synthetic scaling sweep of the materialised ($\bar{A}$ via `torch.sparse.mm`) and matrix-free incidence arms, RTX 3060 Ti / AMD Ryzen 5 5600X, torch 2.6.0+cu118, `float32`. Times are per-repetition medians in ms; build columns cold, the rest warm (protocol in the text).

arm	dev	N	$\mathrm{nnz}(H_u)$	parse	coef	fwd	bwd	epoch	e2e/20	peak (MiB)
mat.	cpu	200	1,600	2.48	10.33	0.143	0.463	0.629	12.42	n/m
mat.	cpu	800	6,400	10.04	40.59	0.513	1.234	1.406	28.01	n/m
mat.	cpu	3,200	25,600	39.83	153.35	2.025	4.421	4.614	91.92	n/m
mat.	cpu	12,800	102,400	160.03	636.68	8.269	21.897	20.560	413.60	n/m
mat.	cpu	51,200	409,600	720.47	2608.31	35.365	86.205	87.612	1764.78	n/m
mat.	gpu	200	1,600	2.51	142.82	0.120	0.632	0.797	15.48	17.47
mat.	gpu	800	6,400	10.01	171.19	0.120	0.620	0.788	15.29	18.50
mat.	gpu	3,200	25,600	41.92	281.36	0.153	0.687	0.834	16.24	26.54
mat.	gpu	12,800	102,400	155.01	705.92	0.361	1.666	1.669	32.07	52.37
mat.	gpu	51,200	409,600	710.05	2533.78	1.170	6.157	6.174	122.09	162.74
inc.	cpu	200	1,600	2.60	8.10	0.133	0.458	0.624	12.34	n/m
inc.	cpu	800	6,400	10.48	32.10	0.236	0.748	0.931	19.40	n/m
inc.	cpu	3,200	25,600	40.70	124.13	1.674	2.656	3.417	50.80	n/m
inc.	cpu	12,800	102,400	165.23	495.11	6.993	11.107	14.417	198.79	n/m
inc.	cpu	51,200	409,600	725.53	1940.13	39.210	68.905	70.020	1393.04	n/m
inc.	gpu	200	1,600	2.53	118.66	0.171	0.594	0.751	14.51	17.77
inc.	gpu	800	6,400	10.07	143.75	0.173	0.621	0.789	15.30	19.29
inc.	gpu	3,200	25,600	40.11	244.00	0.179	0.637	0.798	15.60	27.84
inc.	gpu	12,800	102,400	163.66	611.29	0.435	1.204	1.214	22.86	62.56
inc.	gpu	51,200	409,600	708.69	2098.34	1.468	4.358	4.371	86.12	201.43

Table 8: The coefficients of (5) at the cardinalities the released datasets actually produce, $c \in \{2, 3, 4\}$, for representative orders M. Every cell is generated from (2)–(5) in exact rational arithmetic by the function `ab_weights_exact` (supplementary material); none is transcribed. $M = 81$ is `House`'s full-scale order (Table 5); the row is the exact-rational coefficient port, not a claim that the released signal einsum (which supports $M \leq 5$) runs at this order — there both exact values are ratios of $\sim$40-digit integers, so only their rounding is printed. The last column is the row-mass identity (14), which holds exactly at every cell (checked for all $2 \leq c \leq M \leq 89$).

c	M	$a_{M,c}$ (diagonal)	$b_{M,c}$ (off-diagonal)	$a_{M,c} + (c-1)b_{M,c}$
2	2	0	1	1
2	3	$1/3 \approx 0.3333$	$2/3 \approx 0.6667$	1
2	5	$7/15 \approx 0.4667$	$8/15 \approx 0.5333$	1
2	10	$255/511 \approx 0.4990$	$256/511 \approx 0.5010$	1
2	81	≈ 0.5000	≈ 0.5000	1
3	3	0	$1/2$	1
3	5	$6/25 = 0.24$	$19/50 = 0.38$	1
3	10	$605/1866 \approx 0.3242$	$1261/3732 \approx 0.3379$	1
3	81	≈ 0.3333	≈ 0.3333	1
4	4	0	$1/3$	1
4	5	$1/10 = 0.1$	$3/10 = 0.3$	1
4	10	$1554/6821 \approx 0.2278$	$5267/20463 \approx 0.2574$	1
4	81	≈ 0.2500	≈ 0.2500	1

agree exactly. Deduplicating to H_u before applying Theorem 1 reproduces the assignment semantics exactly: our sparse build matches the dense reference array entry for entry (max absolute difference 0.000e+00 on every dataset and both normalisation settings).

For $E = \{\{1, 2\}, \{1, 2\}\}$ ($M = 2$), $\bar{A}_0$ counts the node-set once but $D = 2I$ counts it twice, so the rows of $D^{\dagger}\bar{A}_0$ sum to $\frac{1}{2}$. Counting a repeated node-set in both numerator and normaliser, or in neither, gives row-stochastic rows instead, and the incidence form supports either. The closed form assumes a binary incidence matrix with nonempty columns and $M \geq 2$; the reference builder reads only the support of each column, so weighted hyperedges are not representable in it.

E.6 THE COEFFICIENTS

As M grows, padding dominates and both weights of Table 8 approach $1/c$, e.g. $a_{M,2} = (2^{M-2} - 1)/(2^{M-1} - 1) \to \frac{1}{2}$. At House's $M = 81$ every pair weight is within 2% of $1/c$, so off the diagonal the layer is close to a plain clique expansion.

F TWO-SECTION INDISTINGUISHABILITY AND THE PASCH WITNESS

F.1 PROOF OF THEOREM 3

Proof. By (9) the model sees the hypergraph only through $\widetilde{A}_H$, and every stage commutes with reordering rows by a permutation P. Row-wise maps do: the encoder gives $x_A = Px_B$ (right-multiplication by W_0, the shared bias, since $P\mathbf{1} = \mathbf{1}$, and the entrywise ReLU all commute with P), and $g(PY) = Pg(Y)$. The channel sums ignore row order, $S(x_A) = \mathbf{1}^\top Px_B = S(x_B)$, so $\Sigma = \mathrm{diag}(S)^{M-2}$ is shared (the exponent because $M_A = M_B$). Propagation carries P through: $\widetilde{A}_{H_A} x_A \Sigma = P\widetilde{A}_{H_B} P^\top P x_B \Sigma = P\,\widetilde{A}_{H_B} x_B \Sigma$. Hence $F_\theta(H_A, X_A^{(0)}) = P\,F_\theta(H_B, X_B^{(0)})$ for every θ. In the normalised case, $(PD_BP^\top)^\dagger = PD_B^\dagger P^\top$, so (11) and $\bar{A}_{0,A} = P\bar{A}_{0,B}P^\top$ give $D_A^\dagger \bar{A}_{0,A} = P\,D_B^\dagger \bar{A}_{0,B}\,P^\top$, which is (10). The Pasch pair has $\bar{A}_A = \bar{A}_B$ and $D_A = D_B = 2I$, so $P = I$. □

Remark 4 (Scope of the hypotheses)**.** The theorem requires: common order M; equality of the *effective* operator $\widetilde{A}_H$ (i.e. $D^\dagger \bar{A}$ in the normalised case) under a node permutation P; correspondingly permuted X_0; and shared *deterministic* row-wise maps in g (encoder $\to S \to$ propagation $\to$ readout). For the normalised operator, equal $\bar{A}$ together with equal raw degree under P is sufficient for the effective-operator equality; equal $D^\dagger \bar{A}$ is the direct condition. Dropout is not deterministic: samplewise equality needs paired permuted masks, otherwise the statement is equality in distribution. A model with arbitrary node-ID-dependent readouts is outside this theorem.

F.2 THE PASCH WITNESS

Witness (Pasch trade). The 3-uniform sextets $\mathcal{H}_A = \{123, 145, 246, 356\}$ and $\mathcal{H}_B = \{124, 135, 236, 456\}$ have the same degree sequence (2 everywhere) and the same 12-pair multiset, so by Theorem 1 $\bar{A}_A = \bar{A}_B$ exactly. But the node completing the anchor pair $\{1, 2\}$ is 3 under $\mathcal{H}_A$ and 4 under $\mathcal{H}_B$. The paired Pasch task presents each copy of $\mathcal{H}_A$ together with a twin copy of $\mathcal{H}_B$, marks the anchors in X and labels that completion; its two query nodes then carry bit-identical features and bit-identical $\bar{A}X$ rows but opposite labels, pinning any $g(\bar{A}X)$ model to exactly 50%.

In the task, the correspondence P is the identity on the six node labels, the features are two anchor indicators, a query indicator shared by nodes 3 and 4, and a constant column, and $M = 3$. Rows 3 and 4 of $\bar{A}$ are identical, and since $D = 2I$ for both hypergraphs the same holds for $D^\dagger \bar{A}$.

As uncoloured hypergraphs, $\mathcal{H}_A$ and $\mathcal{H}_B$ are isomorphic: the transposition $(3\,4)$ maps one onto the other. The witness is therefore not a pair from two isomorphism classes but a labelled incidence trade under the identity correspondence, two labellings of one class that share $\bar{A}$ and X exactly while disagreeing on which labelled node completes the anchor pair.

What the 50% *proves.* For a balanced pair of observations that a deterministic model of the form (9) maps to the same representation while they carry opposite labels, exactly one of the two is classified correctly on every draw. This is a ceiling for that model family under the hypotheses of Theorem 3, not for hypergraph networks in general: EHNN (Kim et al., 2022) and the k-GWL/k-HNN hierarchy (Zhang et al., 2025) are not of the form (9). The Pasch pair is the smallest labelled instance of the graph trades of Aksoy et al. (2024, §4.2.3).

F.3 LEARNED SEPARATION

A *world* is one data set of the paired task: 448 twin pairs, each an independently relabelled copy of $\mathcal{H}_A$ and of $\mathcal{H}_B$, split $128/64/256$ into training, validation and test pairs, with twins in the same split. Each copy is its own six-node hypergraph and copies never interact: T-Spectral (exact) and the clique-graph GCN propagate within a copy, the set-based models only along hyperedges. Each model uses one configuration selected on two development worlds (width 32, learning rate 3×10^{-4},

Table 9: Learned separation on the paired Pasch task: test accuracy (%), paired t_9 95% interval over 10 world means (3 initialisations each, averaged within world), epoch selected by validation accuracy. Chance is 50%. Small type gives the identifier used in the code. The printed (symmetrised) layer is the T-Spectral (exact) arm with its operator replaced by (12).

arm	test acc	95% CI (t_9)	macro-F1
MLP, features only `mlp`	50.00	[50.00, 50.00]	33.3
clique-graph GCN `kgraph_l1`	50.00	[50.00, 50.00]	33.3
T-Spectral (exact), primary `tcmp_l1`	50.00	[50.00, 50.00]	33.3
AllDeepSets `alldeepsets`	**100.00**	[100.00, 100.00]	100.0
AllSetTransformer `allsettransformer`	**100.00**	[100.00, 100.00]	100.0
ED-HNN `edhnn`	**100.00**	[100.00, 100.00]	100.0
printed (symmetrised) layer `tspec_sym_l1`	**100.00**	[100.00, 100.00]	100.0

no dropout) and is confirmed on 10 held-out worlds $\times$ 3 initialisations. Initialisations are averaged within each world, so the paired t_9 interval of Table 9 is over 10 independent world means. A graph-solvability control checks that each model can be trained. Its label depends only on pairwise co-occurrence; every structured model reaches 100% on it and the feature-only MLP stays at 50%. The 50% of T-Spectral (exact) on the main task is therefore the bound of the theorem, not a training failure.

The printed (symmetrised) layer runs under the same protocol, as the T-Spectral (exact) arm with the operator $\bar{A}x$ replaced by (12), and S_c summed over the six nodes of each copy. It reaches 100% on the main task and on the graph-solvability control. Its margins are small: the largest paired logit difference is about one percent of the logit scale, since at $M = 3$ the higher-order block is damped by $1/N$ with $N = 6$.

G Parity and numerical stability

G.1 Parity

Table 10 compares the exact evaluator of (4) with a from-definition dense oracle on the five reference hypergraphs, at their order. *fp64-direct* agrees to 10^{-15}–10^{-16}; the 10^{-7}–10^{-8} residual of *complex64-reference* is the reference implementation's single-precision complex cast. All 750 comparisons (five datasets, ten seeds, three widths, five evaluation routes; forward, loss and gradients of one pass) pass. A separate run covers both normalisation settings, with a worst single-precision forward residual of 7.2×10^{-7}.

Table 10: Parity of the exact evaluator against a from-definition dense oracle, one row per released fixture family, unnormalised setting. Relative max-norm residuals (scaled by the reference norm, floor 10^{-30}), reported separately for the two independent comparisons: *fp64-direct* (both sides in `float64`; tolerances 10^{-10} forward/loss, 10^{-9} gradients) and *complex64-reference* (a literal reconstruction of the reference implementation's internal `complex64` cast; tolerances 10^{-6} forward/loss, 10^{-5} gradients). All 750/750 checks pass.

	fp64-direct				complex64-reference			
dataset	fwd	loss	∇_x	∇_W	fwd	loss	∇_x	∇_W
`coauth./cora`	3.53e-16	4.02e-16	3.97e-16	2.62e-16	1.02e-07	1.86e-08	7.96e-08	2.48e-08
`coauth./dblp`	6.02e-16	3.22e-16	4.03e-16	2.79e-16	1.19e-07	5.16e-08	7.24e-08	5.72e-08
`cocit./cora`	4.80e-16	3.40e-16	3.84e-16	4.05e-16	1.69e-07	4.47e-08	6.97e-08	4.09e-08
`cocit./citeseer`	4.23e-16	3.70e-16	5.59e-16	3.17e-16	1.20e-07	4.89e-08	1.51e-07	3.61e-08
`cocit./pubmed`	4.25e-16	2.65e-16	5.23e-16	4.05e-16	1.37e-07	3.04e-08	6.98e-08	4.23e-08
worst, any cell	**6.02e-16**	**4.02e-16**	**5.59e-16**	**4.05e-16**	**1.69e-07**	**5.16e-08**	**1.51e-07**	**5.72e-08**

Table 11: Large-scale parity on edge-seeded fixtures (fp64-direct / complex64-reference, as in Table 10, same tolerances). "Worst" is the maximum relative residual over all scored quantities, seeds and widths, including the optimiser trajectories. Uniform random-node sampling (subsets of at most 400 nodes) formed no usable fixture, and the dense oracle's size guard refused both full datasets.

dataset	fixtures	orders	fp64 worst	c64-ref. worst
`new/smallWalmart`	20	$M = 5$	8.94e-16	3.19e-07
`new/House`	5	$M \in \{3, 4, 5\}$	3.54e-15	7.14e-07

G.2 Large-scale parity

The dense oracle cannot be built for `House` and `smallWalmart` (Table 5), and uniform random node subsets small enough for it contain no complete hyperedge. Fixtures are instead seeded on one hyperedge each and induced on its node set: 20 from `smallWalmart`, all at $M = 5$, and 5 from `House`, seeded on its small hyperedges ($M \in \{3, 4, 5\}$). All 25 pass the full battery of Table 10 at the same tolerances, adding 1- and 10-step optimiser trajectories and node-relabelling and edge-order invariance, 3,750 scored comparisons in all (Table 11). The full-size passes are timed in Table 6.

G.3 Numerical stability of the tube-power factor

The factor S_c^{M-2} of (4) overflows single precision once $(M-2)\log_{10} S_{\max} > 38.5$; at the House order $M = 81$ this happens as soon as a nonnegative channel sum exceeds about 3.07. With $T := M - 2$, the log-domain reading

$$\log|Z[:,c]| = \log|\bar{A}x[:,c]| + T\log|S_c|, \qquad \operatorname{sign} Z[:,c] = \operatorname{sign}\big(\bar{A}x[:,c]\big)\cdot \operatorname{sign}(S_c)^T, \tag{15}$$

with a per-channel shift subtracted before any exp, never forms the power. A channel with $S_c = 0$ is set to exact zero for $T \geq 1$, and the factor is 1 at $T = 0$; under the released encoder $S_c \geq 0$ and the parity factor is $+1$.

We vary $M \in \{2, \ldots, 6, 10, 40, 81\}$, N from 10 to 3×10^5 (degree surrogate at large N; N^M is never allocated), hyperedge size, feature scale 10^{-2}–10^2, hidden width and precision (single, double, mixed), with at least 3 seeds per cell. In single precision the linear-domain forward pass is finite in 75% of the cells: in every cell with $M \leq 6$, but in only 29% at $M = 81$. The log-domain forward pass, loss and gradient norm are finite in all 5,760 cells, including $M = 81$ at $N = 3 \times 10^5$. Where both readings are finite, they agree to 3×10^{-13} relative in double precision and to 10^{-4} in single precision, and deleting the parity factor $\operatorname{sign}(S_c)^T$ produces 976 sign mismatches, so the check detects a wrong sign rule. The identity is exact at every order, but its literal linear form is not a usable forward map at the orders the census reaches. The evaluator therefore works in the log domain, or absorbs the factor into W_1 (Remark 1); T-Spectral (exact) is trained the second way.

H The printed (symmetrised) layer

The printed (symmetrised) layer of Wang et al. (2025, §III-B–C) symmetrises before the convolution. Each tube axis of both the adjacency tensor and the cross-node interaction signal is reflection-padded to length $2N+1$ as $[0,\ v,\ \mathrm{reverse}(v)]$, with a factor 2^{-T} per padded input, where $T = M - 2$. The t-product is then taken circularly over the padded axes. In the final layer "an inverse symmetrization operation is first applied", which keeps tube positions $1, \ldots, N$ of each padded axis and rescales by 2^T, and only after that does the readout sum the N^{M-2} frontal slices (their eq. 15). The reference implementation defines the padding and its inverse (`symmetric()`, `anti_symmetric_torch()`) but does not call them, so the released layer is the unsymmetrised readout of Proposition 1. Here we give the closed form of the one-layer printed (symmetrised) layer.

Closed form (inverse symmetrisation, then sum). Write $S_c = \sum_m x_{mc}$. The two padded tubes convolve independently along each axis, and the length-N window that inverse symmetrisation keeps turns each tube axis into the factor $g_c[k] = S_c - \frac{1}{2}x_{kc}$: the two index-range partial sums

that make up the window telescope. Expanding the product over the T axes in elementary symmetric polynomials gives (12) of §5 for any tensor A. There $B_0 = \bar{A}x$, so the $s = 0$ term is exactly the collapse $Z = \bar{A}x\,\mathrm{diag}(S)^{M-2}$ of (4), and $B_T = Y$ with $Y[i,c] = \sum_{j,a} A[i,j,a]\,x_{jc} \prod_{r=1}^{T} x_{a_r c}$, since e_T is the product of its arguments: the T-Spatial contraction (eq. 17 of the original paper), whose $M = 3$ case is the matched sum of §5. Equation (12) agrees with a dense construction of the padded convolution from its definition to $\sim 10^{-16}$ relative at $M \in \{3, 4\}$. Evaluated from the tensor's nonzero tuples, it costs $O(T\,\mathrm{nnz}(A)\,d)$ time: the dense array is never formed, but $\mathrm{nnz}(A)$ grows combinatorially with M (Appendix E.2).

Table 12: The released layer uses all slice pairings, T-Spatial only matched ones, so the two compute different functions. $A^{[k]}$, $X^{[k]}$: tensor and signal slices at tube position k. T-Spatial's "first output slice" (the term of Wang et al. (2025)) is a slice of a slice-wise product, not of the t-product; their eq. 17 writes it as the matched sum in the last row.

	released layer (their eq. 15)	T-Spatial (their eqs. 16–17)
product	t-product: FFT, matrix product per frequency, inverse FFT	slice-wise product, no FFT
readout	sum of all N^{M-2} output slices	first output slice
pairs used	all pairs: $\sum_k \sum_l A^{[k]} X^{[l]}$	matched pairs: $\sum_k A^{[k]} X^{[k]}$

Summing all $(2N{+}1)^T$ padded slices without the inverse symmetrisation, a reading the original paper does not describe, does collapse: the padding doubles each input's total along every axis, so that readout equals $4^{M-2}Z$.

I Additional DHG-Bench results

We run under the protocol of DHG-Bench (Li et al., 2026): its per-seed random $50/25/25$ split, 10 seeds and final-epoch test accuracy; we add macro-F1. **T-Spectral (exact)** is the matrix-free evaluator of Theorem 1, run on DHG-Bench's data and split generator. These are DHG-Bench's graphs, distinct from the five hypergraphs of the reference archive (Table 5) even where a name recurs. The baselines are MLP, HGNN (Feng et al., 2019), CEGCN (a GCN on the clique expansion; Li et al., 2026), UniGCNII (Huang and Yang, 2021), AllSetTransformer (Chien et al., 2022), ED-HNN (Wang et al., 2023), TF-HNN (Tang et al., 2025) and EHNN (Kim et al., 2022), as implemented in DHG-Bench. DHG-Bench's own tensor baseline, T-MPHN, is the message-passing variant, not the layer studied here. Tables 1 (accuracy, in the body) and 13 (macro-F1) give every model at its per-dataset configuration (Appendix J).

Configurations. Each baseline runs at the configuration DHG-Bench ships for the dataset. DHG-Bench ships none for `citeseer`, so there every baseline runs at its generic default. On `walmart-trips-100`, ED-HNN's tuned configuration (hidden width 512) runs out of memory on our 8 GiB GPU, and EHNN was not run because its cache and hypernetwork construction at 88,860 nodes exceeded our compute; neither missing cell is a null result. ED-HNN at its generic default does run there, at 77.92, below the best baseline. T-Spectral (exact) is chosen on validation accuracy (Appendix J), whereas the "best baseline" of the body is the test-best baseline, which favours the baselines.

Paired differences. Paired by seed (seed s of both runs uses the same split), T-Spectral (exact) minus the best baseline, mean and 95% t-interval half-width over the 10 seeds, is $+0.37 \pm 0.65$ on Cora, $+0.63 \pm 0.58$ on Citeseer, $+0.50 \pm 0.20$ on PubMed, -3.78 ± 1.04 on House-c., -1.64 ± 0.25 on Walmart-t. and -3.73 ± 0.33 on Actor. A 95% interval that excludes zero marks a margin that is consistent across seeds; it does not account for the difference in tuning search.

Retuned baselines. TF-HNN and UniGCNII, the strongest baselines on the homophilic sets, are retuned there with the selection rule of our arm. Starting from DHG-Bench's shipped configuration, a grid of 72 configurations varies width $\in \{64, 128\}$, dropout $\in \{0.3, 0.5, 0.7\}$, weight decay $\in \{0, 5 \cdot 10^{-4}, 10^{-2}\}$, learning rate $\in \{10^{-3}, 10^{-2}\}$ and depth $\in \{1, 2\}$ (diffusion hops for TF-HNN, convolutions for UniGCNII), at the configuration's own number of epochs. Every configuration is

Table 13: DHG-Bench node classification, macro-F1 (%), same runs and protocol as Table 1.

	homophilic			heterophilic		
model	Cora	Citeseer†	PubMed	House-c.	Walmart-t.	Actor
MLP	$73.86_{\pm 0.89}$	$67.54_{\pm 1.07}$	$86.50_{\pm 0.25}$	$83.16_{\pm 0.87}$	$46.22_{\pm 0.41}$	$80.03_{\pm 0.66}$
HGNN	$76.43_{\pm 1.02}$	$66.45_{\pm 1.18}$	$85.83_{\pm 0.23}$	$78.75_{\pm 1.37}$	$66.39_{\pm 0.32}$	$66.10_{\pm 0.34}$
CEGCN	$75.40_{\pm 0.55}$	$63.16_{\pm 0.83}$	$85.73_{\pm 0.21}$	$56.61_{\pm 1.66}$	$59.17_{\pm 0.26}$	$49.63_{\pm 0.43}$
UniGCNII	$78.41_{\pm 0.88}$	$68.31_{\pm 1.16}$	$87.33_{\pm 0.21}$	$83.53_{\pm 0.85}$	$34.55_{\pm 0.57}$	$74.93_{\pm 0.73}$
AllSetTransformer	$76.63_{\pm 0.96}$	$63.03_{\pm 0.82}$	$87.50_{\pm 0.21}$	$83.15_{\pm 0.87}$	$69.11_{\pm 0.33}$	$80.26_{\pm 0.58}$
ED-HNN	$76.32_{\pm 0.94}$	$64.91_{\pm 0.86}$	$87.79_{\pm 0.25}$	$83.06_{\pm 0.85}$	OOM	$80.13_{\pm 0.45}$
TF-HNN	$78.28_{\pm 0.76}$	$68.62_{\pm 1.10}$	$87.77_{\pm 0.24}$	$82.56_{\pm 0.78}$	$65.68_{\pm 0.27}$	$80.04_{\pm 0.59}$
EHNN	$73.23_{\pm 1.17}$	$63.65_{\pm 1.11}$	$86.25_{\pm 0.58}$	$63.36_{\pm 7.40}$	—	$81.35_{\pm 0.46}$
T-Spectral (exact)	$78.61_{\pm 0.52}$	$68.76_{\pm 1.18}$	$88.28_{\pm 0.22}$	$79.71_{\pm 1.35}$	$65.26_{\pm 0.30}$	$75.85_{\pm 0.55}$

Table 14: Retuned baselines on the homophilic sets: TF-HNN and UniGCNII at DHG-Bench's shipped configuration (as in Table 1) and retuned over 72 configurations, selected on validation, next to T-Spectral (exact). Final-epoch test accuracy and macro-F1 (%), mean and 95% CI half-width over 10 seeds, protocol of Table 1. †As there.

	accuracy			macro-F1		
model	Cora	Citeseer†	PubMed	Cora	Citeseer†	PubMed
TF-HNN, shipped	$80.03_{\pm 0.82}$	$72.55_{\pm 0.95}$	$88.03_{\pm 0.24}$	$78.28_{\pm 0.76}$	$68.62_{\pm 1.10}$	$87.77_{\pm 0.24}$
TF-HNN, retuned	$79.51_{\pm 0.76}$	$72.69_{\pm 0.86}$	$88.25_{\pm 0.25}$	$78.02_{\pm 0.76}$	$68.84_{\pm 1.07}$	$88.02_{\pm 0.24}$
UniGCNII, shipped	$80.03_{\pm 0.78}$	$73.78_{\pm 0.70}$	$87.52_{\pm 0.20}$	$78.41_{\pm 0.88}$	$68.31_{\pm 1.16}$	$87.33_{\pm 0.21}$
UniGCNII, retuned	$79.82_{\pm 0.97}$	$73.94_{\pm 0.51}$	$88.35_{\pm 0.22}$	$78.18_{\pm 1.06}$	$69.35_{\pm 1.01}$	$88.25_{\pm 0.21}$
T-Spectral (exact)	$80.40_{\pm 0.57}$	$74.41_{\pm 0.90}$	$88.53_{\pm 0.23}$	$78.61_{\pm 0.52}$	$68.76_{\pm 1.18}$	$88.28_{\pm 0.22}$

screened at seeds 0 to 2; the three with the best mean final-epoch validation accuracy are rerun at seeds 0 to 9, and the one with the best ten-seed validation accuracy is reported at its test accuracy (Table 14).

Retuning raises both baselines on Citeseer and PubMed but not on Cora, where validation picks configurations that score below the shipped ones on test. In accuracy T-Spectral (exact) stays ahead of the better retuned baseline on all three sets, by 0.18 to 0.58 points (in macro-F1, both retuned baselines are ahead on Citeseer), so the homophilic ties of §6 persist after retuning these baselines. The other baselines, and every baseline on the heterophilic sets, keep their shipped configurations.

In the shipped-configuration runs of Table 1, UniGCNII's optimiser uses learning rate 10^{-2} and weight decay 10^{-2} on its convolutions and $5 \cdot 10^{-4}$ elsewhere whatever its configuration file sets, as DHG-Bench's own training script does; the grid above sets both for all its parameters.

The inserted normalisation. On PubMed and the three heterophilic sets the selected configuration places a row-wise normalisation between propagation and readout (Table 15); the released layer has none. Removing it from the selected configuration, all else fixed, changes accuracy by -16.65 ± 1.46 on PubMed, -3.10 ± 3.63 on House-c., -10.43 ± 0.29 on Walmart-t. and -1.74 ± 0.32 on Actor (paired by seed, mean and 95% t-interval half-width over the 10 seeds). Rerun unchanged, the selected configurations reproduce their accuracy to within 0.56 points. The drop is outside seed noise on PubMed, Walmart-trips and Actor; on House-committees its interval includes zero. These configurations were selected with normalisation, so removing it alone may overstate its contribution. Re-selected by the same rule from configurations without the normalisation (163–337 per dataset, fewer than with it), T-Spectral (exact) scores 88.00 ± 0.27 on PubMed, 79.81 ± 0.74 on House-c., 77.25 ± 0.14 on Walmart-t. and 81.68 ± 0.31 on Actor (mean and 95% CI half-width over the 10 seeds), that is -0.53, 0.00, $+0.15$ and -1.12 points from Table 1 (differences computed before rounding). So at an equal selection rule the normalisation accounts for at most about one point of Table 1, on Actor.

J DHG-BENCH TUNING AND CONFIGURATIONS

Baselines. The baselines run at DHG-Bench's shipped configurations (Appendix I). For ED-HNN on `cora`, that is dropout 0.5, hidden width 512 and no MLP layers, where the generic default has 0.2, 128 and 2.

T-Spectral (exact). Every selected model uses an encoder, one propagation by $\bar{A}$ or its row-normalised form $D^{-1}\bar{A}$ (both from the reference builder), and a linear readout. The search may add a LayerNorm or RMSNorm between the propagation and the readout. Every run validates every epoch. The first-stage grid (216 one-step configurations) is propagation $\in \{\bar{A}, D^{-1}\bar{A}\}$; normalisation after propagation $\in$ {layer, none, rms}; width $\in \{64, 128\}$; dropout $\in \{0.3, 0.5, 0.7\}$; weight decay $\in \{0, 5 \cdot 10^{-4}, 10^{-2}\}$; learning rate $\in \{10^{-3}, 10^{-2}\}$, at 100 epochs. To fit the compute budget, `walmart-trips-100` fixes learning rate to 10^{-3} (108 configurations). A second stage adds the next value outward on every axis on which one of the five best first-stage configurations (three on `walmart-trips-100`) sits at the grid edge, reaching width $\in \{32, 256\}$; dropout $\in \{0.1, 0.2, 0.8\}$; weight decay $\in \{3 \cdot 10^{-2}, 10^{-1}\}$; learning rate $\in \{3 \cdot 10^{-4}, 3 \cdot 10^{-2}\}$, and trains each candidate for 100 and 200 epochs. It ran on every dataset. A third stage repeats the extension once around the best configurations of the first two stages, reaching width $\in \{512\}$; dropout $\in \{0.9\}$. It added configurations on `cora`, `citeseer`, `pubmed`, `house-committees-100` and `walmart-trips-100`. Selection pools every stage. The first two stages also trained two-step configurations (two propagations by $\bar{A}$); they are not the released layer and are not eligible for selection. Every configuration is screened at seeds 0 to 2. The three configurations with the best mean final-epoch validation accuracy are rerun at seeds 0 to 9, and the one with the best ten-seed validation accuracy is reported at its test accuracy (Table 15). No choice uses test accuracy.

Table 15: Selected configuration of T-Spectral (exact) per dataset, all with one propagation step. "norm": normalisation after propagation, before the linear readout. "configs": configurations the selection chose from, over all stages; "val" and "test": ten-seed mean final-epoch accuracy (%) at the selected configuration; the test column is the T-Spectral (exact) row of Table 1.

dataset	propagation	norm	width	dropout	w. decay	lr	epochs	configs	val	test
Cora	$D^{-1}\bar{A}$	none	512	0.9	0	$3 \cdot 10^{-4}$	200	254	80.72	80.40
Citeseer	$D^{-1}\bar{A}$	none	64	0.8	$5 \cdot 10^{-4}$	10^{-3}	100	291	74.20	74.41
PubMed	$D^{-1}\bar{A}$	rms	512	0.8	10^{-2}	10^{-3}	100	479	88.72	88.53
House-c.	$D^{-1}\bar{A}$	layer	128	0.8	$3 \cdot 10^{-2}$	10^{-2}	200	519	80.00	79.81
Walmart-t.	$D^{-1}\bar{A}$	layer	256	0.3	10^{-2}	10^{-3}	200	273	77.02	77.10
Actor	$\bar{A}$	layer	128	0.5	$5 \cdot 10^{-4}$	10^{-2}	200	299	83.31	82.79

K TRUNCATED ARCHIVE DATASETS

The `ModelNet40` and `NTU` files in the distributed archive are truncated and were excluded. The archive passes its CRC check, so the truncation is present in the distributed files rather than introduced by transfer or unpacking.

L EXTENDED RELATED WORK

The incidence closed form and TTSV2. Theorem 1 is assembled from known ingredients, including the pair $(a_{M,c}, b_{M,c})$ itself. The multiplicity $L_{M,c} = c!\,S(M,c)$ of (2) is the normalisation of Banerjee et al. (2017), which the reference builder writes. The tube sum $\bar{A} = A \cdot \mathbf{1}^{M-2}$ is the all-ones case of TTSV2, the tensor-times-same-vector-in-all-but-two contraction of Aksoy et al. (2024). Specialising their generating functions (supplement, §SM1–SM2) at $T = M - 2$ and edge weight $w_e = c/(c!\,S(M,c))$ gives

$$a_{M,c} = w_e\, T!\,[t^T]\, e^t(e^t-1)^{c-1}, \qquad b_{M,c} = w_e\, T!\,[t^T]\, e^{2t}(e^t-1)^{c-2},$$

which expand to exactly (5); the two derivations agree as exact rationals for $1 \le c \le M \le 8$. Writing a hypergraph operator as $H\,\mathrm{diag}(w)\,H^\top$ plus a diagonal correction is, independently, the clique-expansion normalisation of Zhou et al. (2006) and Agarwal et al. (2006). Our contribution

is the exact identification: the tube-summed tensor of the released t-product layer is exactly this specialisation of TTSV2, with the assignment, deduplication and normalisation semantics of its builder made explicit and preserved. The reference `GENTTSV` implementation at the all-ones vector matches our coefficients to double-precision tolerance on small fixtures of order 3 to 5. Hypergraphs that a weighted clique expansion cannot separate are likewise in print as graph trades (Aksoy et al., 2024, §4.2.3), and Jiang et al. (2026) study the same information loss; what we add is the derivation that makes the released layer inherit it exactly.

M REPRODUCIBILITY MAP

The supplementary `MANIFEST.tsv` maps results and claims to scripts and run records. `REGENERATE.md` gives their reproduction commands, and `ENVIRONMENT.md` records the software and hardware. The package also contains exact-arithmetic nullspace certificates and `proofs/check_examples.py`, which checks the worked examples.