CAPACITY COUNTS: LEARNING TO PREDICT HYPEREDGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Hyperedge prediction scores a candidate node set from observed group interactions. Historical hyperedges may repeatedly rely on the same members, so their count can overstate the support that admits distinct representatives. We introduce CapacityDegree, which represents this assignable support through capacity-constrained matching. A recursive rule assigns node support levels while bounding how many hyperedges each member can represent. For a candidate, matching rank measures available support, and deletion responses identify individual and joint dependencies. A compact readout combines the rank profile with degrees of the pair-response matrix. We prove that capacity levels retain information beyond pairwise projections and satisfy a sharp member-removal bound. We also characterize the deletion responses and prove that the degree branch strictly enlarges the model's function class even when all other inputs, including the rank profile, agree. Across seven datasets, CapacityDegree outperforms eight baselines in AP and AUC, improving macro AP by 9.26 percentage points over the strongest baseline.

1 INTRODUCTION

Many observed interactions involve a set of participants: legislators join a committee, people participate in a contact event, and substances occur in a drug product. A hypergraph records each interaction as a hyperedge, preserving group membership beyond pairwise links (Battiston et al., 2020). Given the observed hypergraph and a candidate node set, hyperedge prediction asks how likely that set is to be an interaction. Higher-order link prediction has been studied through local interaction patterns and learned representations (Benson et al., 2018; Wan et al., 2021). We consider binary scoring of supplied candidates from a fixed node universe.

A candidate can draw support from historical hyperedges containing its members. The number of those hyperedges does not reveal how their support is distributed. For example, six historical hyperedges may all have the same member as their only eligible representative. They provide six units of coverage, but only one assignable unit when a member can represent one hyperedge. More generally, even with the same numbers of hyperedges and eligible members, their incidence pattern determines how many distinct assignments are possible. This motivates a representation of both the amount of assignable support and its dependence on particular members.

CapacityDegree applies this principle at two scales. At the node level, a recursive rule assigns a support level when enough incident hyperedges can be represented by sufficiently supported members. Each hyperedge supplies one unit, and each member has a fixed integer capacity. At the candidate level, we restrict the eligible representatives to its members and examine the resulting matching rank. Deleting a member measures its individual contribution; deleting a pair can expose an additional loss hidden by either deletion alone. The model summarizes these quantities with a rank profile and the weighted degree of the pair-response matrix, then combines them with learned node embeddings.

Our contribution connects recursive support levels to a candidate representation through the same assignment constraint, building on recursive structural descriptions and matching ranks (Batagelj & Zaveršnik, 2002; Chen et al., 2019; Edmonds & Fulkerson, 1965). The analysis establishes what each scale adds. Capacity levels distinguish hypergraphs with identical pairwise projections and count levels, and their change under member removal is bounded sharply by the member capacity.

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.

For candidate representations, matroid duality characterizes the individual and joint deletion losses. We then construct two hypergraphs on which all inputs to the model without the degree branch agree, including the rank profiles, while the degree branch separates them. This proves a strict increase in expressivity for the specified architecture. An exact alternating-path algorithm makes these pair responses computable without a separate max-flow solve for each pair.

The experiments assess predictive performance on seven public datasets against eight baselines, including HyperNoRA and S3Hyper. CapacityDegree achieves the highest AP and AUC on every dataset. Component removals measure the contributions of the profile and degree readouts; parameter-matched replacements examine the effect of using matching information in place of coverage and counts. All comparisons use fixed size-matched negative sets and support graphs that exclude their training targets.

2 Related work

Hypergraph representations. Spectral hypergraph learning uses normalized incidence operators to encode group relations (Zhou et al., 2006). HGNN learns through this type of propagation (Feng et al., 2019), extending the neighborhood aggregation used in graph convolution (Kipf & Welling, 2017). HNHN adds nonlinear hyperedge representations (Dong et al., 2020), AllSet models node and hyperedge aggregation with multiset functions (Chien et al., 2022), and UniGNN relates graph and hypergraph convolutions within a common framework (Huang & Yang, 2021). HyperGCN constructs graph approximations of hyperedges, including a fixed approximation in its fast variant (Yadati et al., 2019). These encoders can be paired with a set readout for candidate scoring.

Set representations range from pooled elementwise maps in Deep Sets (Zaheer et al., 2017) to permutation-averaged functions in Janossy pooling (Murphy et al., 2019) and attention-based interactions in Set Transformer (Lee et al., 2019). The discriminating power of aggregation depends on the structural information presented to it (Xu et al., 2019). CapacityDegree supplies its set readout with matching responses computed from the historical hyperedges and the candidate members.

Hyperedge prediction. Simplicial closure characterizes how observed interactions precede new higher-order links (Benson et al., 2018). HPRA derives scores through resource allocation (Kumar et al., 2020), and local random-walk methods compare node distributions using Jensen–Shannon divergence (Xu et al., 2023). Neural predictors include NHP (Yadati et al., 2020), Hyper-SAGNN's self-attention model for variable-size hyperedges (Zhang et al., 2020), and SNALS's structural and spectral features (Wan et al., 2021). AHP learns a negative sampler for hyperedge prediction (Hwang et al., 2022), while HyperSearch searches for new hyperedges over unconstrained candidate sets (Choo et al., 2025). Our task is scoring supplied candidates under a common sampling protocol.

Self-supervised hypergraph learning can contrast nodes, hyperedges, and their memberships across augmented views, as in TriCL (Lee & Shin, 2023). HyperNoRA combines node relations with self-supervision for hyperedge prediction (Li et al., 2026b); S3Hyper combines candidate substructures with self-supervised objectives (Li et al., 2026a). Both are included in our experiments. Hypergraph symmetry breaking provides another route to expressive higher-order link prediction (Zhang et al., 2024). CapacityDegree derives candidate features from assignments to historical hyperedges and the support lost when candidate members are removed.

Structural patterns and assignable support. Graph motifs reveal organization beyond individual edges (Benson et al., 2016), while hypergraph motifs describe overlap patterns among connected hyperedges (Lee et al., 2020). Recursive descriptions provide a complementary view: generalized cores replace degree with more general vertex properties (Batagelj & Zaveršnik, 2002), and D-chain tomography introduces a parameterized structural spectrum for graphs (Chen et al., 2019). Hypergraph core definitions include minimum node and hyperedge degrees (Lee et al., 2023) and rules that allow hyperedges to survive partial member removal (Bu et al., 2023).

Our recursive rule bounds how often each member can represent an incident hyperedge. Distinct representatives are characterized by Hall's matching condition (Hall, 1935), and the associated local matching rank is a transversal-matroid rank (Edmonds & Fulkerson, 1965). We use these classical tools to derive capacity levels and candidate pair responses. The resulting model connects recursive support levels to a compact candidate representation.

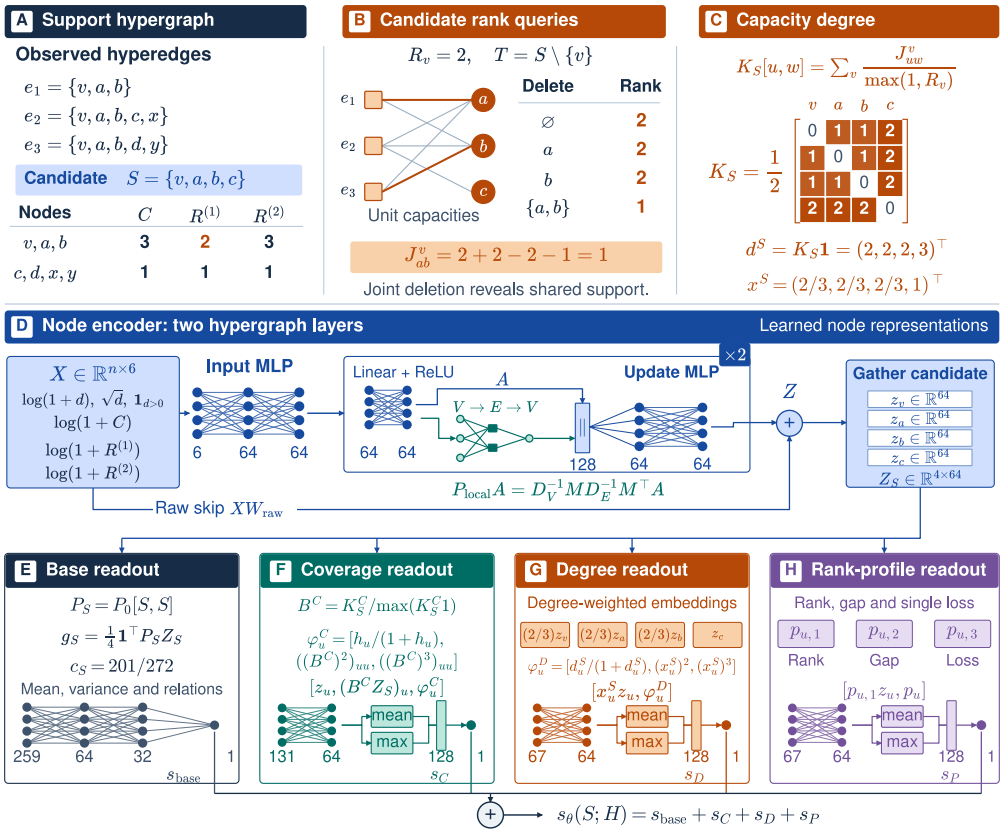


Figure 1: CapacityDegree: a worked support example and the prediction architecture. (A–C) Support levels, a root matching query, and the aggregated degree response for the same candidate. (D) Two incidence layers and a raw-feature skip produce node embeddings. (E–H) Ordinary relations, coverage, degree, and the matching-rank profile feed four additive readouts.

3 CapacityDegree

CapacityDegree first computes support levels on the observed hypergraph, then uses them in node encoding and candidate matching queries. The readout combines ordinary relations, coverage, and the rank and deletion features of those queries (Figure 1).

3.1 Support levels

Let $H = (V, \mathcal{E})$ be a finite, unweighted support hypergraph, with no duplicate hyperedges and $|e| \geq 2$. Let $d_v = |\{e \in \mathcal{E} : v \in e\}|$. A candidate is a set $S \subseteq V$, $|S| \geq 2$, scored without inserting it into H. All constructions in this section use only H and S.

For a nonnegative integer vector x, a level k, and a lag $a \geq 0$, define

$$U_{ve}(x, k) = \{u \in e \setminus \{v\} : x_u \geq k - a\}. \tag{1}$$

Only incident hyperedges with $U_{ve}(x, k) \neq \varnothing$ are eligible. The count rule $G_a(x)_v$ is the largest $k \leq d_v$ for which at least k incident hyperedges are eligible. Descending iteration from d gives its greatest fixed point C_a.

The capacity rule forms a bipartite graph whose left vertices are these incident hyperedges and whose right vertices are their eligible members. Each hyperedge has capacity one; each member has capacity $b \in \{1, 2, \ldots\}$. An integral assignment chooses at most one representative per hyperedge and uses each member at most b times. If $\nu_b(v; x, k)$ is its maximum size, define

$$F_{a,b}(x)_v = \max\big(\{0\} \cup \{k \leq d_v : \nu_b(v; x, k) \geq k\}\big). \tag{2}$$

Descending iteration gives the greatest fixed point $R_{a,b}$. Each root has its own assignment problem, with member capacity shared across its incident hyperedges. We use $a = 1$ and $b \in \{1, 2\}$ throughout the experiments, writing $C = C_1$ and $R^{(b)} = R_{1,b}$. Eligibility requires at least one qualifying member besides the root; we refer to this as the two-member rule.

The maps are monotone, the iterations terminate, and $R_{a,b} \le C_a$. Also, $R_{a,b}$ is nondecreasing in b. Appendix A.1 gives the proofs. We compute exact integral levels with max flow (Ford & Fulkerson, 1956), retaining distinct historical hyperedges as separate resources even when their eligible member sets coincide.

3.2 Node encoding and relation readout

We form six node features by standardizing the three degree coordinates $\log(1 + d_v)$, $\sqrt{d_v}$, and $\mathbf{1}[d_v > 0]$, and the three coordinates $\log(1+C_v)$, $\log(1+R_v^{(1)})$, and $\log(1+R_v^{(2)})$. Standardization uses all known node IDs within each support context; no node labels or raw attributes are used. Let X be the feature matrix and M the incidence matrix. The node encoder uses the ordinary support operator

$$P_{\text{local}} = D_V^{-1} M D_E^{-1} M^\top, \qquad D_V = \text{diag}(d), \quad D_E = \text{diag}(|e|), \tag{3}$$

where the inverse degree is zero at isolated nodes. Starting with $h^{(0)} = \text{MLP}_{\text{in}}(X)$, two layers compute

$$A^{(\ell)} = \text{ReLU}(h^{(\ell)} W_\ell + b_\ell), \qquad h^{(\ell+1)} = \text{MLP}_\ell([A^{(\ell)}, P_{\text{local}} A^{(\ell)}]), \qquad Z = h^{(2)} + X W_{\text{raw}}. \tag{4}$$

All hidden widths are 64. The layer output is an update, with a separate raw-feature skip at the end.

For candidate relations, remove the diagonal of P_{local} and normalize each nonzero row:

$$(P_0)_{vu} = \frac{\mathbf{1}[u \neq v](P_{\text{local}})_{vu}}{\sum_{w \neq v} (P_{\text{local}})_{vw}}, \tag{5}$$

with a zero row when the denominator vanishes. This relation matrix shares the observed incidence structure with the encoder.

For $s = |S|$, the base readout uses the mean μ_S and coordinatewise variance σ_S^2 of the embeddings, together with

$$g_S = \frac{1}{s} \sum_{v,u \in S} (P_0)_{vu} z_u, \qquad c_S = \frac{1}{s} \sum_{v,u \in S} (P_0)_{vu}. \tag{6}$$

We do not renormalize the submatrix on S. Thus c_S retains the amount of relation mass inside the candidate. The base logit is

$$s_{\text{base}}(S) = \text{MLP}_{\text{base}}([\mu_S, \sigma_S^2, g_S, c_S, \log(1 + s)]). \tag{7}$$

3.3 Candidate capacity degree

The relation readout summarizes node pairs. The matching query asks how a candidate's members can jointly represent historical support. For root $v \in S$, let $T = S \setminus \{v\}$ and $R_v = R_v^{(1)}$. A member u is eligible for an incident hyperedge e when $u \in e \setminus \{v\}$ and $R_u^{(1)} \ge R_v - 1$. Let $\nu_v(U)$ be the maximum unit-capacity matching between members in $U \subseteq T$ and these historical hyperedges. Define the capped rank

$$f_v(U) = \min\{R_v, \nu_v(U)\}. \tag{8}$$

Deletion queries keep the support levels and historical resource identities fixed. For distinct $u, w \in T$, the joint response is

$$J_{uw}^v = f_v(T \setminus \{u\}) + f_v(T \setminus \{w\}) - f_v(T) - f_v(T \setminus \{u, w\}), \qquad J_{uu}^v = 0. \tag{9}$$

It is the loss caused by deleting both members minus the sum of the two individual losses. A positive value therefore identifies a loss hidden by separate deletions. For example, if three members can represent two interchangeable support resources, deleting either one preserves rank two; deleting two reduces the rank to one.

Appendix Figure 3 illustrates two support systems with equal full rank and equal single-deletion losses, but different pair-deletion responses.

Aggregate over the candidate roots and take row sums:

$$K_S[u,w] = \sum_{v \in S \setminus \{u,w\}} \frac{J^v_{uw}}{\max(1, R_v)}, \qquad d^S_u = \sum_{w \in S} K_S[u,w]. \tag{10}$$

The capacity degree d^S_u is the weighted degree in the candidate response matrix: it measures the total pair-deletion response involving u across roots. Let $x^S_u = d^S_u / \max_w d^S_w$, with all $x^S_u = 0$ when all degrees vanish. The three structural inputs are

$$\phi^D_u = \left[\frac{d^S_u}{1 + d^S_u}, (x^S_u)^2, (x^S_u)^3 \right]. \tag{11}$$

For nodewise vectors t_u, write $\mathrm{pool}(t) = [\mathrm{mean}_u\, t_u, \max_u t_u]$. The degree branch is

$$t^D_u = \mathrm{ReLU}(W_D[x^S_u z_u, \phi^D_u]), \qquad s_D(S) = w_D^\top \mathrm{pool}(t^D). \tag{12}$$

Both maps are bias-free, so a zero response matrix gives an exactly zero degree-branch logit. The branch combines degree-weighted embeddings through nodewise transformations and pooling.

Coverage branch. Coverage supplies a complementary input. Replace $f_v(U)$ by $\min\{R_v, |N_v(U)|\}$, where $N_v(U)$ is the set of historical hyperedges touched by eligible members in U. This counts covered edges without enforcing one edge per member. Apply Eqs. (9)–(10) to obtain K^C_S. With $h = K^C_S \mathbf{1}$, set $B^C = K^C_S / \max_u h_u$ (zero when the denominator vanishes) and

$$\phi^C_u = \left[\frac{h_u}{1 + h_u}, ((B^C)^2)_{uu}, ((B^C)^3)_{uu} \right], \quad t^C_u = \mathrm{ReLU}(W_C[z_u, (B^C Z_S)_u, \phi^C_u] + b_C). \tag{13}$$

A linear mean/max readout gives $s_C(S)$.

3.4 Matching rank and single-member losses

Pair responses describe joint losses; they do not determine the full rank or the individual losses. Write $T_v = S \setminus \{v\}$, $D_v = \max(1, R_v)$, and $c_v(U) = \min\{R_v, |N_v(U)|\}$. For member $u \in T_v$, let

$$\delta_v(u) = f_v(T_v) - f_v(T_v \setminus \{u\}).$$

We attach three coordinates to each candidate member:

$$p_v = \left[\frac{f_v(T_v)}{D_v}, \frac{c_v(T_v) - f_v(T_v)}{D_v}, \frac{1}{s-1} \sum_{q \in S \setminus \{v\}} \frac{\delta_q(v)}{D_q} \right]. \tag{14}$$

They measure normalized assignable support, the gap between coverage and assignment, and the member's average single-deletion loss across the other roots. All three lie in $[0, 1]$. The profile readout is

$$t^P_v = \mathrm{ReLU}(W_P[p_{v,1} z_v, p_v]), \qquad s_P(S) = w_P^\top \mathrm{pool}(t^P). \tag{15}$$

Both maps are bias-free. The full logit is

$$s_\theta(S; H) = s_{\mathrm{base}}(S) + s_C(S) + s_D(S) + s_P(S). \tag{16}$$

We learn the neural parameters with binary cross-entropy on the candidate labels. The model has 74,306 parameters; support levels, relation matrices, and candidate structural features are fixed throughout training.

3.5 Computation

For one root query with t candidate peers and A active member–resource incidences, an initial matching and alternating-path repairs compute all pair responses in $O(tA+t^2)$ time. The profile uses a matching and its single-deletion indicators in $O(tA + t)$ time. Costs are summed over candidate roots. Coverage preprocessing forms dense matrix powers in $O(s^3)$ time, and candidate response matrices occupy $O(s^2)$ space. These structural quantities are cached; degree and profile evaluation then require nodewise transformations and pooling. Appendix B gives the exact query algorithm and implementation details.

4 STRUCTURAL PROPERTIES

The analysis follows the representation from support levels to candidate scores: capacity constraints preserve information lost by pairwise counts, and joint deletion responses add information beyond ranks and individual losses. All results use the two-member eligibility rule. Proofs are in Appendix A.

When capacity limits support. Fix a root v, levels x, and threshold k. Let $\mathcal{L}$ be its eligible incident hyperedges, $m = |\mathcal{L}|$, and $N(A) = \bigcup_{e \in A} U_{ve}(x, k)$ for $A \subseteq \mathcal{L}$. Max-flow/min-cut (Ford & Fulkerson, 1956) gives

$$\nu_b(v; x, k) = m - \Delta_b, \qquad \Delta_b = \max_{A \subseteq \mathcal{L}} \big\{ |A| - b|N(A)| \big\}. \tag{17}$$

The empty set makes $\Delta_b \geq 0$. A level is feasible exactly when $\Delta_b \leq m - k$: the largest assignment deficit must fit within the excess number of eligible edges. Thus a subset of edges can block the level even when total edge and member counts are sufficient. At $x = C_a$, the condition holds at $k = C_{a,v}$ for every root if and only if $R_{a,b} = C_a$ (Appendix A.2).

Theorem 1 (Information beyond a pairwise projection)**.** *There are two simple 3-uniform hypergraphs on the same 12 nodes with identical incidence Gram matrices $MM^\top$, identical count levels C_1, and identical P_{local}, but different unit-capacity levels $R_{1,1}$ at a designated root: 6 in one graph and 8 in the other.*

The construction assigns every pair of ten peers to a triple containing either the root or an auxiliary node. The two assignments preserve all pairwise counts. Their root support systems have different maximum matchings. Thus the capacity levels encode assignability that the pairwise projection and count features jointly fail to determine.

Theorem 2 (Bound under removal of one member)**.** *Fix $a \geq 0$ and integer $b \geq 1$. Remove a member w from all incident hyperedges, retain each surviving hyperedge's identity, and discard sets of size less than two. Denote the result by $H \ominus w$, allowing distinct surviving edges with the same member set. For every remaining node v,*

$$0 \leq R^H_{a,b}(v) - R^{H \ominus w}_{a,b}(v) \leq b. \tag{18}$$

The upper bound is attained for every b. The analogous loss for the count rule can be arbitrarily large.

The bound follows because at most b assignment units can use the removed member. Lowering every remaining level by b, clipped at zero, preserves feasibility of the remaining representatives. The capacity b therefore controls the maximum support-level loss under the stated member-removal operation.

Proposition 3 (Pair responses from a dual matroid)**.** *For a fixed root and candidate peer set, f_v is the rank function of a truncated transversal matroid. Each off-diagonal entry of J^v is either zero or one. It equals one exactly when the two elements are nonloops in the same parallel class of the dual matroid. Thus J^v is a disjoint union of clique adjacency matrices and isolated vertices.*

This characterization follows from matroid rank and duality. Let $\mathcal{P}_v$ be the nonloop parallel classes of the dual matroid at root v. Summing their clique degrees gives the exact identity

$$d^S_u = \sum_{v \in S \setminus \{u\}} \frac{1}{\max(1, R_v)} \sum_{P \in \mathcal{P}_v} \mathbf{1}[u \in P](|P| - 1). \tag{19}$$

Each root contributes the number of other members in u's class, scaled by $1/\max(1, R_v)$. Thus the degree has a direct structural meaning: it counts a member's partners in joint support loss across roots. Since every contribution is nonnegative, $d^S = 0$ if and only if $K_S = 0$ (Appendix A.6).

Proposition 4 (Deletion losses at different orders)**.** *Fix a root and write $T = T_v$, $f = f_v$. Let f^* be the dual rank on T. For every $A \subseteq T$,*

$$f(T) - f(T \setminus A) = |A| - f^*(A). \tag{20}$$

For a pair, this loss is $\delta_v(u) + \delta_v(w) + J^v_{uw}$. Moreover, $\delta_v(u) = 1$ implies $J^v_{uw} = 0$ for every w. For integer $k \geq 0$, if the uncapped matching rank is at least $R_v + k$, removing any k or fewer members leaves the capped rank unchanged.

Thus individually necessary members and nonzero pair responses represent different deletion behavior. The profile records the former; the degree records the latter. A zero pair response does not imply zero assignable support. For example, four members each reaching four interchangeable resources, with cap two, have rank two and zero single and pair losses; deleting three members loses one rank unit. More generally, Eq. (20) locates joint losses in dependent sets of the dual. Appendix A.7 proves these statements and gives equal-coverage examples where zero pair responses accompany different ranks.

Theorem 5 (Information added by the degree branch)**.** *There are two simple hypergraphs on the same 28 nodes, each with 171 hyperedges, and a common unobserved seven-node candidate S, such that X, P_{local}, P_0, K_S^C, and all profiles p_v agree exactly. Every model comprising the encoder, base, coverage, and profile branches therefore gives the same score on both inputs for every parameter choice at inference. Their capacity response matrices are $K_S = 0$ and $K_S = M_0/6$, where M_0 is the adjacency matrix of two disjoint three-node cliques and the isolated candidate root. The degree branch can assign scores 0 and $3/14$. Hence adding this branch strictly enlarges the specified model's function class.*

The proof uses two families of four support groups on six peers:

$$(123, 145, 246, 356) \qquad \text{and} \qquad (123, 123, 456, 456). \tag{21}$$

Both have rank four, and each single deletion preserves that rank. Only the second has nonzero pair-deletion responses. A construction with a second root, resource markers, and external anchors preserves all the other network inputs. In the second candidate, the six peer degrees are $1/3$, so the first coordinate in Eq. (11) is $1/4$. Selecting that coordinate and mean pooling gives $6/(7\cdot 4) = 3/14$. The construction demonstrates why aggregating pair responses can add information even when all relation, coverage, and profile inputs agree.

5 Experiments

We evaluate complete predictors, the contributions of the two matching readouts, and the effects of replacing matching quantities with coverage and counts.

5.1 Datasets and protocol

We evaluate on HouseCommittees, SenateBills, EmailEnron, ContactPrimarySchool, ContactHighSchool, NDCSubstances, and NDCClasses from the public collection of Benson (n.d.). Repeated node sets are collapsed, singleton hyperedges are removed, and all remaining incidences have unit weight. Timestamps and node labels are unused. The known node universe is retained, including IDs that become isolated after filtering. Appendix D gives exact processed sizes and split counts.

We split unique positive hyperedges into 60% training, 20% validation, and 20% test with split seed 42. Each positive has one negative of the same size. Negative members are sampled without replacement from active training nodes with probabilities proportional to training degree raised to 0.75. All known positives are used solely to reject false negatives. Negative sets are fixed and disjoint across splits.

Training positives are shuffled with fold seed 20260915 and alternated into two target folds. Each fold is scored using the opposite fold as its support graph. Its associated negatives use that same graph. The target positive is therefore absent from its support graph. A single parameter set is shared across both contexts. At validation and test time, we average the two context logits before computing metrics. Both support contexts remain fixed for validation and test.

We use hidden width 64, dropout 0.3, and Adam (Kingma & Ba, 2015) with learning rate 10^{-3} and weight decay 10^{-4}. All main comparisons use the same fixed training budget and validation-AP checkpoint selection. We report test average precision (AP) and ROC-AUC for initialization seeds 42 and 43 on the same fixed partition. Appendix C gives the exact budgets and remaining optimization details.

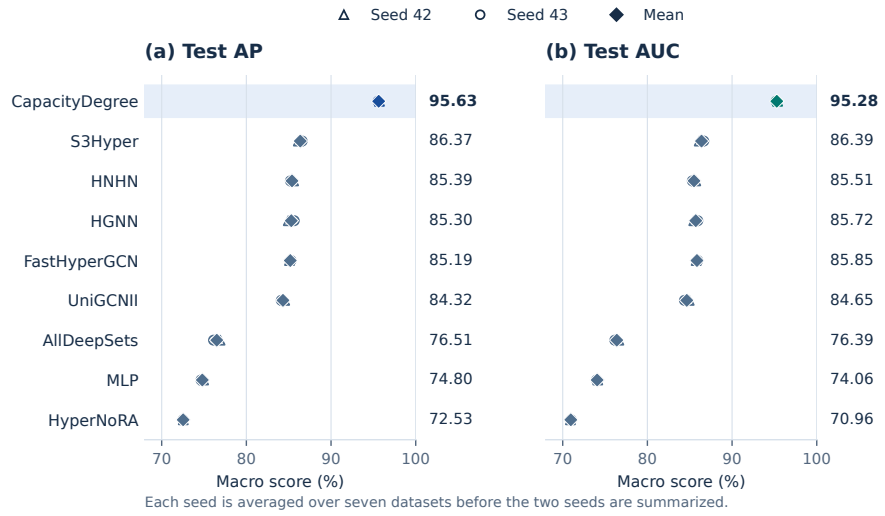


Figure 2: Macro test AP and AUC across seven datasets. Triangles and circles show initialization seeds 42 and 43; diamonds show their mean. Segments span the two seed scores.

Table 1: Test AP (%), averaged over initialization seeds 42 and 43. Mean weights the seven datasets equally; bold marks the highest mean. Primary/High denote the contact datasets; NDC-S/C denote substances/classes.

Model	House	Senate	Enron	Primary	High	NDC-S	NDC-C	Mean
CapacityDegree	**99.260**	**97.305**	**92.137**	**91.062**	**91.467**	**98.386**	**99.787**	**95.629**
MLP	82.951	75.329	65.598	63.703	58.032	82.280	95.674	74.795
HGNN	97.124	88.453	80.534	71.952	70.614	91.197	97.259	85.304
HNHN	97.531	88.835	79.725	68.445	70.685	95.454	97.046	85.389
AllDeepSets	83.715	74.699	65.023	63.632	57.248	94.365	96.863	76.507
UniGCNII	94.060	86.516	78.332	70.912	69.366	93.830	97.247	84.323
FastHyperGCN	85.822	87.896	82.422	75.501	77.151	90.357	97.151	85.186
HyperNoRA	73.242	68.372	68.829	67.057	67.930	77.740	84.575	72.535
S3Hyper	71.678	94.070	86.987	80.784	81.917	90.628	98.493	86.365

5.2 Baseline comparisons

We compare eight baselines. MLP, HGNN, HNHN, AllDeepSets, UniGCNII, and FastHyperGCN use a shared mean-embedding and size head. HyperNoRA (Li et al., 2026b) retains its node-relation readout and contrastive objective; S3Hyper (Li et al., 2026a) retains its substructure-aware attention readout and three self-supervised objectives. All eight use three standardized degree features and the same split, negative candidates, support contexts, hidden width, optimizer, and main training budget. These comparisons evaluate complete predictors with their respective structural inputs, readouts, and objectives. Appendix E specifies each implementation.

CapacityDegree achieves the highest AP and AUC on all seven datasets. Its macro AP is 95.63% and AUC is 95.28%. It exceeds S3Hyper, the strongest baseline by both macro metrics, by 9.26 AP and 8.89 AUC percentage points.

Full AUC and sample standard deviations are in Appendix F. Figure 2 compares macro scores and initialization variability; Appendix Figure 5 shows gains over each dataset's strongest baseline.

Table 2: Component ablations: macro test scores across five datasets. ΔAP is full minus ablated AP in percentage points. All rows use the same ablation training budget.

Component	AP (%)	AUC (%)	ΔAP
Full model	96.000 ± 0.068	95.751 ± 0.114	+0.000
w/o profile	95.343 ± 0.075	95.020 ± 0.080	+0.657
w/o degree	96.005 ± 0.009	95.734 ± 0.039	-0.005

Table 3: Parameter-matched mechanism comparisons on two datasets. AP and AUC are equal-weight macro test percentages.

Model	Parameters	AP (%)	AUC (%)
CapacityDegree	74,306	98.282 ± 0.029	98.194 ± 0.025
Coverage profile	74,306	98.184 ± 0.058	97.998 ± 0.140
Count + coverage profile	74,306	98.249 ± 0.159	98.160 ± 0.284

5.3 Component ablations

We separately remove the profile or degree branch, retaining capacity-level node features and the other readout. Both variants and a full reference use the same ablation budget. We report equal-weight means over five datasets and sample standard deviations across two seed-level macro scores. Appendix F.1 lists the datasets.

Removing the rank profile reduces macro AP by 0.657 percentage points and AUC by 0.731 points, with decreases in both seed-level macro scores. Removing the degree branch leaves the macro scores nearly unchanged: AP rises by 0.005 points and AUC falls by 0.017 points.

5.4 Matching and coverage controls

Two parameter-matched replacements on HouseCommittees and SenateBills compare matching features with coverage and counts. The coverage-profile control retains the exact degree branch and replaces the profile's ranks and single-deletion losses with capped coverage quantities. It isolates the choice of profile information. The count control additionally replaces capacity-level node features and degree responses with their count or coverage counterparts, assessing the combined capacity representation. Both controls have 74,306 parameters and use the main training budget and the same two seeds.

At the same parameter count, CapacityDegree exceeds the coverage-profile control by 0.098 AP and 0.196 AUC percentage points. Its gains over the count-and-coverage profile are 0.034 points on both metrics. The full model therefore has higher two-dataset means than both matched controls.

Appendix F.2 gives the other relation-readout and mechanism replacements, with their parameter counts and results.

6 Conclusion

CapacityDegree represents historical support through assignments constrained by member capacity. The same principle defines recursive node levels and candidate matching queries; rank and deletion responses describe the amount of support and its dependencies. The analysis separates capacity levels from pairwise projections, gives a sharp member-removal bound, and proves that the degree branch adds expressive power beyond the rank profile. On seven datasets, CapacityDegree leads all eight baselines in AP and AUC. Component ablations identify a predictive contribution from the rank profile, linking assignable support and individual deletion losses to hyperedge prediction.

AI USE STATEMENT

Generative AI tools assisted with writing, visualization, implementation, and mathematical analysis. The authors checked the final claims, proofs, code, and experimental records and take responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

Section 3 defines the model, and Appendix C specifies the remaining architecture and numerical details. Appendix A provides the proofs and explicit constructions. The dataset preprocessing, support split, negative sampling, checkpoint selection, and optimizer are described in Section 5 and Appendix D. Appendix F reports per-dataset metrics and initialization variability.

ETHICS STATEMENT

This study uses public interaction datasets and does not collect new personal data. It scores sampled candidate sets without inferring demographic attributes. Scores from a balanced sampled benchmark are not calibrated probabilities of real-world interaction. Any use to infer private social or medical relations would require a separate assessment of data permissions and potential harm.

REFERENCES

Vladimir Batagelj and Matjaž Zaveršnik. Generalized cores. arXiv:cs/0202039, 2002. URL https://arxiv.org/abs/cs/0202039.

Federico Battiston, Giulia Cencetti, Iacopo Iacopini, Vito Latora, Maxime Lucas, Alice Patania, Jean-Gabriel Young, and Giovanni Petri. Networks beyond pairwise interactions: Structure and dynamics. *Physics Reports*, 874:1–92, 2020. doi: 10.1016/j.physrep.2020.05.004. URL https://arxiv.org/abs/2006.01764.

Austin R. Benson. Datasets. Cornell University dataset collection, n.d. URL https://www.cs.cornell.edu/~arb/data/. Accessed September 17, 2026.

Austin R. Benson, David F. Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016. doi: 10.1126/science.aad9029. URL https://arxiv.org/abs/1612.08447.

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115 (48):E11221–E11230, 2018. doi: 10.1073/pnas.1800683115. URL https://arxiv.org/abs/1802.06916.

Claude Berge. Two theorems in graph theory. *Proceedings of the National Academy of Sciences*, 43(9):842–844, 1957. doi: 10.1073/pnas.43.9.842. URL https://doi.org/10.1073/pnas.43.9.842.

Fanchen Bu, Geon Lee, and Kijung Shin. Hypercore decomposition for non-fragile hyperedges: Concepts, algorithms, observations, and applications. *Data Mining and Knowledge Discovery*, 2023. doi: 10.1007/s10618-023-00956-2. URL https://arxiv.org/abs/2301.08440.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1725–1735, 2020. URL https://proceedings.mlr.press/v119/chen20v.html.

Ricky X. F. Chen, Christian M. Reidys, and Andrei C. Bura. D-chain tomography of networks: A new structure spectrum and an application to the SIR process. *SIAM Journal on Applied Dynamical Systems*, 18(4):2181–2201, 2019. doi: 10.1137/19M1243890. URL https://arxiv.org/abs/1810.07004.

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*, 2022. URL https://arxiv.org/abs/2106.13264.

Philip S. Chodrow, Nate Veldt, and Austin R. Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021. doi: 10.1126/sciadv.abh1303. URL https://arxiv.org/abs/2101.09611.

Hyunjin Choo, Fanchen Bu, Hyunjin Hwang, Young-Gyu Yoon, and Kijung Shin. HyperSearch: Prediction of new hyperedges through unconstrained yet efficient search. arXiv:2510.17153, 2025. URL https://arxiv.org/abs/2510.17153.

Yihe Dong, Will Sawin, and Yoshua Bengio. HNHN: Hypergraph networks with hyperedge neurons. arXiv:2006.12278, 2020. URL https://arxiv.org/abs/2006.12278.

Jack Edmonds and D. R. Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards, Section B*, 69B(3):147–153, 1965. URL https://nvlpubs.nist.gov/nistpubs/jres/69B/jresv69Bn3p147_A1b.pdf.

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, pp. 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558. URL https://ojs.aaai.org/index.php/AAAI/article/view/4235.

L. R. Ford, Jr. and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi: 10.4153/CJM-1956-045-5. URL https://doi.org/10.4153/CJM-1956-045-5.

P. Hall. On representatives of subsets. *Journal of the London Mathematical Society*, s1-10(1):26–30, 1935. doi: 10.1112/jlms/s1-10.37.26. URL https://doi.org/10.1112/jlms/s1-10.37.26.

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*, pp. 2563–2569, 2021. doi: 10.24963/ijcai.2021/353. URL https://www.ijcai.org/proceedings/2021/353.

Hyunjin Hwang, Seungwoo Lee, Chanyoung Park, and Kijung Shin. AHP: Learning to negative sample for hyperedge prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022. URL https://arxiv.org/abs/2204.06353.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Tarun Kumar, K Darwin, Srinivasan Parthasarathy, and Balaraman Ravindran. HPRA: Hyperedge prediction using resource allocation. In *Proceedings of the 12th ACM Conference on Web Science*, 2020. doi: 10.1145/3394231.3397903. URL https://arxiv.org/abs/2006.11070.

Dongjin Lee and Kijung Shin. I'm me, we're us, and I'm us: Tri-directional contrastive learning on hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. URL https://arxiv.org/abs/2206.04739.

Geon Lee, Jihoon Ko, and Kijung Shin. Hypergraph motifs: Concepts, algorithms, and discoveries. *Proceedings of the VLDB Endowment*, 13(12):2256–2269, 2020. doi: 10.14778/3407790.3407823. URL https://arxiv.org/abs/2003.01853.

Jongshin Lee, Kwang-Il Goh, Deok-Sun Lee, and B. Kahng. (k, q)-core decomposition of hypergraphs. *Chaos, Solitons & Fractals*, 173:113645, 2023. doi: 10.1016/j.chaos.2023.113645. URL https://arxiv.org/abs/2301.06712.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, 2019. URL https://proceedings.mlr.press/v97/lee19d.html.

Ming Li, Huiting Wang, Yuting Chen, Lu Bai, Lixin Cui, Feilong Cao, and Ke Lv. Self-supervised hypergraph learning with substructure awareness for hyperedge prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23055–23062, 2026a. doi: 10.1609/aaai.v40i27.39471. URL https://ojs.aaai.org/index.php/AAAI/article/view/39471.

Ming Li, Zhanle Zhu, Xinyi Li, Lu Bai, Lixin Cui, Feilong Cao, and Ke Lv. HyperNoRA: Hyperedge prediction via node-level relation-aware self-supervised hypergraph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23047–23054, 2026b. doi: 10.1609/aaai.v40i27.39470. URL https://ojs.aaai.org/index.php/AAAI/article/view/39470.

Rossana Mastrandrea, Julie Fournet, and Alain Barrat. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PLOS ONE*, 10(9):e0136497, 2015. doi: 10.1371/journal.pone.0136497. URL https://doi.org/10.1371/journal.pone.0136497.

Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1811.01900.

Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):e23176, 2011. doi: 10.1371/journal.pone.0023176. URL https://doi.org/10.1371/journal.pone.0023176.

Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955. doi: 10.2140/pjm.1955.5.285. URL https://msp.org/pjm/1955/5-2/p11.xhtml.

Changlin Wan, Muhan Zhang, Wei Hao, Sha Cao, Pan Li, and Chi Zhang. Principled hyperedge prediction with structural spectral features and neural networks. arXiv:2106.04292, 2021. URL https://arxiv.org/abs/2106.04292.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.00826.

Xin-Jian Xu, Chong Deng, and Li-Jie Zhang. Hyperlink prediction via local random walks and Jensen–Shannon divergence. *Journal of Statistical Mechanics: Theory and Experiment*, 2023:033402, 2023. doi: 10.1088/1742-5468/acc31e. URL https://arxiv.org/abs/2303.14996.

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*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/1efa39bcaec6f3900149160693694536-Abstract.html.

Naganand Yadati, Vikram Nitin, Madhav Nimishakavi, Prateek Yadav, Anand Louis, and Partha Talukdar. NHP: Neural hypergraph link prediction. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, 2020. doi: 10.1145/3340531.

3411870. URL https://malllabiisc.github.io/publications/papers/nhp_cikm20.pdf.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1703.06114.

Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1911.02613.

Simon Zhang, Cheng Xin, and Tamal K. Dey. Expressive higher-order link prediction through hypergraph symmetry breaking. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2402.11339.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems*, volume 19, 2006. URL https://proceedings.neurips.cc/paper/2006/hash/dff8e9c2ac33381546d96deea9922999-Abstract.html.

A PROOFS AND CONSTRUCTIONS

A.1 FIXED-POINT PROPERTIES

The fixed-point argument is a finite instance of monotone lattice theory (Tarski, 1955). All orders on vectors are coordinatewise. Increasing x in Eq. (1) can only add eligible member–resource incidences, so it cannot decrease either the eligible-edge count or a maximum matching. Therefore G_a and $F_{a,b}$ are monotone. They map the finite integer box $[0, d]$ to itself. Starting from $x^{(0)} = d$ gives $x^{(1)} \le x^{(0)}$; monotonicity then gives $x^{(i+1)} \le x^{(i)}$ for every i. The sequence stabilizes after finitely many strict coordinate decreases. Its limit is a fixed point.

If $y \le F_{a,b}(y)$, then $y \le d = x^{(0)}$, and induction gives $y \le x^{(i)}$ for every i. Thus the limit dominates all such vectors, including all fixed points. This proves the greatest-fixed-point characterization. Since a feasible matching of size k uses k distinct eligible hyperedges, $F_{a,b}(x) \le G_a(x)$. Iterating yields $R_{a,b} \le C_a$. Likewise $F_{a,b}(x) \le F_{a,b'}(x)$ for $b \le b'$, and $R_{a,b} \le R_{a,b'}$.

A.2 ASSIGNMENT DEFICITS AND EQUALITY OF SUPPORT LEVELS

Fix v, x, k and retain the identities of the m eligible incident hyperedges in $\mathcal{L}$, including those with equal eligible member sets. Construct a directed network with a source, one vertex per eligible hyperedge, one vertex per eligible member, and a sink. Source-to-edge arcs have capacity one, each eligible edge-to-member arc has capacity $m + 1$, and member-to-sink arcs have capacity b. Integral flows correspond exactly to assignments with at most one representative per edge and at most b assignments per member.

A minimum cut has capacity at most m, so it cannot cut an edge-to-member arc. If $A \subseteq \mathcal{L}$ is the set of edge vertices on the source side, all members in $N(A)$ must also be on that side. Including any further members can only increase the cut capacity. The minimum cut with this choice of A therefore has capacity $m - |A| + b|N(A)|$. Max-flow/min-cut and integrality (Ford & Fulkerson, 1956) give

$$\nu_b(v; x, k) = \min_{A \subseteq \mathcal{L}} \{m - |A| + b|N(A)|\} = m - \max_{A \subseteq \mathcal{L}} \{|A| - b|N(A)|\}.$$

This also covers $m = 0$. It proves Eq. (17) and the feasibility condition $\Delta_b \le m - k$. For $k = m$, the condition reduces to $|A| \le b|N(A)|$ for every A, which is Hall's condition applied after making b copies of each member (Hall, 1935). For example, six eligible edges may collectively reach six members while five of the edges reach only three of them. At $b = 1$, that five-edge subset has deficit at least two, so at most four of the six edges can receive distinct representatives.

For the fixed-point statement, write $c = C_a$. At each root evaluate m_v and $\Delta_{b,v}$ at $x = c, k = c_v$. If $\Delta_{b,v} \le m_v - c_v$ for every v, then $F_{a,b}(c) \ge c$. The count rule gives $F_{a,b}(c) \le G_a(c) = c$, hence c is a capacity fixed point. Since $R_{a,b}$ is the greatest such point and $R_{a,b} \le C_a$, equality follows. Conversely, if $R_{a,b} = c$, feasibility at each root implies the deficit inequalities. Passing the test at a single root alone does not establish equality of that coordinate after the coupled iteration. □

A.3 PROOF OF THEOREM 1

Use vertices $0, 1, \ldots, 10, 11$, where 0 is the root and 11 is auxiliary. For a set L of pairs of peers $\{1, \ldots, 10\}$, form exactly one hyperedge for each peer pair $\{i, j\}$: use $\{0, i, j\}$ if the pair belongs to L and $\{11, i, j\}$ otherwise. Define

$$L_A = \binom{\{1,2,3,4\}}{2} \cup \{\{5,6\},\{7,8\}\},$$
$$L_B = \{12, 23, 34, 14, 15, 26, 37, 48\}.$$

Here and below, a digit string denotes its member set. Each L has eight pairs and the same labeled degree sequence $(3, 3, 3, 3, 1, 1, 1, 1, 0, 0)$. Hence the root–peer and auxiliary–peer co-occurrence counts agree. Every peer pair occurs together exactly once. The root and auxiliary node never co-occur. The incidence degrees are 8 at the root, 9 at every peer, and 37 at node 11. This proves equality of $MM^\top$. Uniform edge size three then gives equality of P_{local}.

For the count rule with $a = 1$, the greatest fixed point is $(8, 9, \ldots, 9, 10)$ in both graphs. These levels are feasible: every peer can use its nine incident hyperedges via the other peer, the root can use its eight edges, and node 11 can count at least ten edges through peers at level 9. The root and peers already meet their degree upper bounds. Node 11 cannot reach 11, since every possible representative has degree at most 9 and would need level at least 10.

For unit capacity, every peer has nine distinct other peers, each on a different incident hyperedge. They provide a matching of size nine at the same levels. Node 11 has a matching of size ten in each complement graph. Explicit choices of its ten representing pairs, assigned respectively to members 1 through 10, are

$$\begin{aligned} A: &\quad 15, 25, 35, 45, 57, 67, 79, 89, 9\,10, 1\,10, \\ B: &\quad 13, 24, 35, 45, 56, 67, 78, 89, 9\,10, 1\,10. \end{aligned}$$

Each list uses ten distinct pairs outside its corresponding L. At the root, the six resources of the K_4 component in L_A have only four members, and its other two resources contribute one each. The maximum matching size is therefore six, and six is feasible. For L_B, assign the four pendant pairs $15, 26, 37, 48$ to members 5–8 and the four cycle pairs to members 1–4. This gives eight. Since these are absolute root matching bounds even without thresholds, the displayed feasible vectors are the greatest fixed points. Thus the root levels are 6 and 8. □

A.4 Proof and tightness of Theorem 2

Let $r = R^H_{a,b}$, and let $y_v = \max\{r_v - b, 0\}$ for $v \neq w$. Fix a root v with $r_v > b$. Choose a feasible matching of r_v support units at level r_v in H. At most b of these units are represented by w. Delete those units and retain the others. There are at least $r_v - b$ remaining units, each represented by some $u \neq w$. Its hyperedge survives with both v and u; resource identity is unchanged. Moreover,

$$r_u \geq r_v - a \quad \Longrightarrow \quad y_u = \max\{r_u - b, 0\} \geq r_v - b - a = y_v - a.$$

These representatives remain eligible at level y_v. The edge and member capacities still hold. If $r_v \leq b$, the zero level is feasible without a witness. Thus $y \leq F^{H\ominus w}_{a,b}(y)$, and the greatest-fixed-point property proves $R^{H\ominus w}_{a,b} \geq y$. Conversely, any assignment after removal is an assignment in the original graph. Extending a feasible post-removal level vector by zero at w makes it a postfixed vector in the original graph, which proves $R^{H\ominus w}_{a,b} \leq r$ on surviving nodes.

For tightness, set $a = 0$, choose $m \geq b$, and use a root v, a member w, a set U of $m + 1$ members, and leaves $l_1, \ldots, l_m$. Include all binary edges of the clique on $\{w\} \cup U$, the triples $\{v, w, l_i\}$, and the edge $\{v\} \cup U$. Clique members have levels at least $m+1$, and every leaf has degree one. At root level at least two, the triples can only use w, so at most b triples can be represented; the last edge adds one. The root level is exactly $b + 1$. After removing w, the triples become $\{v, l_i\}$ and cannot support level two. Only $\{v\} \cup U$ remains eligible, so the root level is one. The loss is b. The count rule has root level $m + 1$ before removal and one after removal. Its loss m has no bound depending only on one removed member. □

The operation $H \ominus w$ retains each historical interaction among its remaining members. When two edges shrink to the same set, their resource identities remain distinct. Deleting whole interactions or coalescing equal surviving sets removes additional resources and defines a different perturbation.

A.5 Proof of Proposition 3

On the peer ground set T, sets of members admitting distinct historical representatives form a transversal matroid (Edmonds & Fulkerson, 1965). Capping its rank at the integer R_v is matroid truncation. Let $f = f_v$ and let f^* be the dual rank. By the dual-rank formula,

$$f^*(A) = |A| - f(T) + f(T \setminus A).$$

Substitution into Eq. (9) gives

$$J^v_{uw} = f^*(\{u\}) + f^*(\{w\}) - f^*(\{u, w\}).$$

If one element is a loop in the dual, its singleton rank is zero and adding it changes no rank, so the expression is zero. For two nonloops both singleton ranks are one. Their pair rank is one or

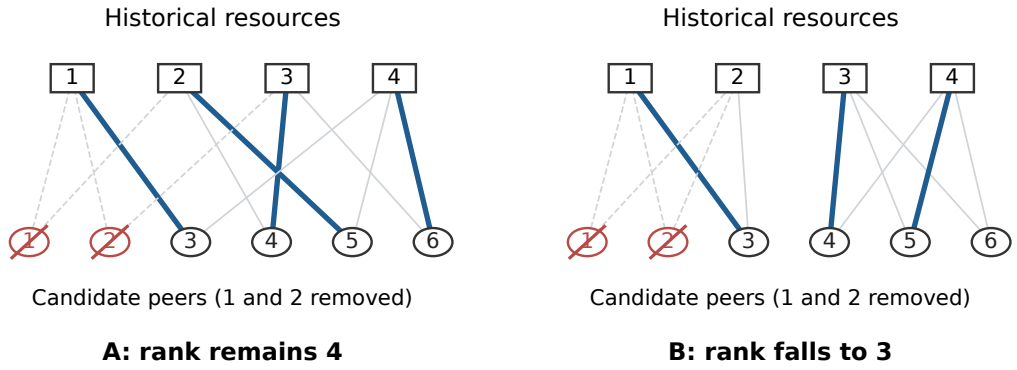


Figure 3: Pair deletion reveals support constraints hidden by single deletions. Both systems have four historical resources and six peers, with unit capacity on each side. Each has rank four, and removing any one peer preserves that rank. After removing peers 1 and 2, system A still admits four assignments, whereas B admits only three. Blue edges show remaining maximum matchings. Thus $J_{12} = 0$ for A and $J_{12} = 1$ for B when the rank cap is at least four.

two, giving response one or zero. Pair rank one defines membership in the same parallel class. Parallelism among nonloops is an equivalence relation. This proves the clique description and the binary response values. If f is additive, direct substitution in Eq. (9) gives zero. □

The coverage branch uses a capped coverage function. Submodularity gives nonnegative joint responses; the binary clique characterization in Proposition 3 is specific to the matching rank.

A.6 Degree decomposition over the rootwise parallel classes

For each root $v \in S$, the nonloop parallel classes $\mathcal{P}_v$ partition the nonloops of the dual matroid on $T = S \setminus \{v\}$. By Proposition 3, a row belonging to $u \in P$ contains one in exactly the other $|P| - 1$ positions of its class; a dual loop has an all-zero row. Therefore

$$\sum_{w \in S \setminus \{u,v\}} J^v_{uw} = \sum_{P \in \mathcal{P}_v} \mathbf{1}[u \in P](|P| - 1).$$

Interchanging the finite sums over w and v in Eq. (10) gives Eq. (19). Every J^v is nonnegative and every root weight $1/\max(1, R_v)$ is positive. Hence K_S is entrywise nonnegative, and its row-sum vector vanishes exactly when K_S vanishes. Equivalently, every root has only singleton nonloop parallel classes. This statement includes zero budgets: if $R_v = 0$, the capped rank is zero and its dual is free, so it has no nontrivial parallel class. When $d^S = 0$, both x^S and ϕ^D vanish, and the two bias-free maps in Eq. (12) give $s_D(S) = 0$. □

A.7 Proof of Proposition 4

For a matroid of rank $f(T)$ on T, dual rank is

$$f^*(A) = |A| - f(T) + f(T \setminus A).$$

Rearrangement gives Eq. (20). For a singleton, $\delta_v(u) = 1 - f^*(\{u\}) \in \{0, 1\}$. For a pair, substitution of Eq. (9) gives

$$f(T) - f(T \setminus \{u, w\}) = \delta_v(u) + \delta_v(w) + J^v_{uw}.$$

If $\delta_v(u) = 1$, then u is a dual loop. Hence $f^*(\{u, w\}) = f^*(\{w\})$, and the joint response is zero for every w. This proves that a necessary individual representative contributes no additional pair-deletion loss at that root.

Let r be the uncapped matching rank. Removing q members destroys at most q edges of any fixed maximum matching, so the surviving rank is at least $r - q$. If $r \geq R_v + k$ and $q \leq k$, both the original

and surviving capped ranks equal R_v. This gives a sufficient condition for unchanged capped rank despite member removal.

Two further examples clarify the information retained by the profile. Take four resources and four candidate members with cap four. In one system, member i reaches only resource i. In another, only member 1 is eligible and reaches all four resources. Both cover four resources and have zero pair responses. Their ranks are four and one, respectively, and their single-deletion vectors are $(1, 1, 1, 1)$ and $(1, 0, 0, 0)$. Thus even coverage together with all pair responses does not determine rank or individual losses. The root budget can be realized independently using eligible members outside the candidate.

Conversely, connect each of four members to all four resources and set cap two. Every set of at least two members has capped rank two. Single and pair deletions lose nothing, while any three-member deletion loses one unit. Its dual is the uniform rank-two matroid on four elements, whose pairs are independent but whose triples are dependent. In general, a deletion set causes loss exactly when it is dependent in the dual, with loss equal to its nullity. The readout uses the full rank, aggregated single-deletion losses, and degree summaries of pair responses. □

A.8 Proof of Theorem 5

Let $P = \{1, \dots, 6\}$, introduce roots r, s, markers $h_1, \dots, h_4$, and an anchor set Q of size 16. Write $A = (123, 145, 246, 356)$ and $B = (123, 123, 456, 456)$. In H_A, include the eight hyperedges

$$\{r, h_j\} \cup A_j, \qquad \{s, h_j\} \cup B_j, \qquad j = 1, \dots, 4.$$

In H_B, exchange A and B between the roots. The common hyperedges are all 15 binary edges on P, all 120 binary edges on Q, four binary edges from each peer to the first four anchors, and two binary edges from each root to the first two anchors. There are 28 vertices and $8+15+120+24+4 = 171$ hyperedges. Markers keep repeated support groups as distinct five-member hyperedges, so both hypergraphs are simple. The candidate $S = \{r\} \cup P$ is not an observed hyperedge.

Pair statistics and levels. Each peer appears twice in each family. Root–peer counts are therefore unchanged by the exchange, as are root–marker counts. Peer–peer counts receive the same sum of contributions from A and B. The co-occurrence of h_j and a peer is the sum of its membership indicators in A_j and B_j. All other edges are common. Thus the entire labeled incidence Gram matrix agrees. Since exchanged edges all have size five, the size-weighted Gram matrix $MD_E^{-1}M^\top$ also agrees, giving the same P_{local}.

Count levels are 6 at each root, 13 at each peer, 2 at each marker, and 15 at each anchor. These values are feasible. Roots, peers, and markers meet degree upper bounds. An anchor level of 16 would require representatives at level at least 15, so only the other 15 anchors qualify and such a level is impossible.

For unit capacity, a peer has five independent binary supports through the other peers and four through anchors. Hence it reaches level nine. Roots and markers have degree at most six and two, so neither can qualify for a peer at any level at least nine. There are only nine remaining eligible neighbors; the peer level is exactly nine. Each root matches its four group resources to distinct peers and its two binary resources to anchors, attaining level six. Markers reach two, and anchors reach 15 through their anchor clique. This gives the stated $R^{(1)}$.

For any integer $b \geq 1$, peer levels remain between 9 and 13, so roots and markers remain ineligible in their queries. After removing those ineligible incidences, every peer sees the same support-group multiset in both hypergraphs: the contributions of both A and B are present regardless of their root assignment. Their greatest-fixed-point subsystem is identical. Root, marker, and anchor levels remain 6, 2, and 15. In particular, $R^{(2)}$ agrees between the graphs. Its peer values are $(12, 12, 12, 13, 13, 13)$; this further detail is not needed for the equality argument. Degree, C, and both capacity vectors agree, so standardized X agrees.

Ordinary relations. Equation (5) is a deterministic row normalization of the common P_{local} after diagonal removal. Thus P_0 agrees exactly between the graphs, as do both repeated relation blocks in the base input.

Candidate responses. At root r, the candidate exposes the four resources of A or B, with the same cap six. Each family has matching rank four, and deleting a single peer preserves rank four. In A, peers can be viewed as the six edges of a K_4 and resources as its four vertices. Any four of these edges have a matching to the four vertices: every subfamily of one, two, three, or four edges touches at least as many vertices. Hence deletion of any two peers preserves rank and $J_A = 0$. In B, peers $1, 2, 3$ share two resources, as do $4, 5, 6$. Deleting two peers from one of these triples loses one rank unit; deleting across triples loses none. Thus J_B consists of two three-node cliques.

At any peer root, the candidate root r is ineligible. Each remaining peer has its own distinct common binary support edge, so the rank is exactly the number of eligible peers retained. It is additive and has zero pair response. Consequently only root r contributes to K_S, with divisor six.

For coverage at root r, each resource contains three candidate peers. Removing at most two peers leaves all four resources covered, so its coverage response is zero in both graphs. At peer roots the eligible support multisets agree, so their entire coverage response matrices agree. Therefore K_S^C, its normalized matrix, and its structural features agree.

Matching-rank profiles. At r, full matching rank and coverage are both four and the budget is six. No candidate peer loses rank individually at r. At a peer root, the five other peers each have a distinct binary support, while r is ineligible. The full rank is five. Coverage is nine: five binary resources and the four group resources containing this peer. The budget is nine. Deleting any one of the five eligible peers loses exactly one matching unit; deleting r loses none. It follows that in both hypergraphs

$$p_r = (2/3, 0, 0), \qquad p_u = (5/9, 4/9, 5/54) \quad (u \in P).$$

The third peer coordinate sums five losses weighted by $1/9$ and then divides by the six other candidate roots. The entire profile agrees, including its coupling to the common node embeddings.

Network outputs. The common encoder and readouts are deterministic functions of these equal inputs at inference, so their outputs agree at every shared parameter setting. Set the base, coverage, and profile output maps to zero. For the degree branch, $d^S = 0$ in H_A and $d_u^S = 1/3$ at the six peers in H_B, with zero at r. Choose one row of W_D to select the first coordinate of ϕ^D, zero all other weights, and let w_D select its mean-pooling coordinate. The outputs are 0 and $3/14$. Finally, setting $w_D = 0$ embeds every model without the degree branch into the full class. The separating function is outside that smaller class, proving strict containment for the stated input map and architecture. □

B EXACT CANDIDATE QUERY ALGORITHM

For one root, store a bipartite adjacency list from the t candidate peers to the historical resources that they can represent. Resource IDs remain distinct even when their member sets are equal. Let A count active incidences, r be the uncapped maximum matching size, and R the frozen root budget. A maximum matching costs $O(tA + t)$ using augmenting paths (Berge, 1957), including isolated peers. A *coloop* is a peer whose deletion lowers the uncapped rank by one. In a maximum matching, the coloops are the matched peers unreachable by alternating paths from unmatched peers.

If $R = 0$, every response is zero. Otherwise the following cases suffice:

1. If $r \geq R + 2$, two deletions cannot lower the uncapped rank below R. Every response is zero.
2. If $r = R + 1$, single deletions never lower the capped rank. A pair lowers it exactly when both members are coloops in the uncapped matroid. Thus the response is the off-diagonal outer product of the coloop indicators.
3. If $r \leq R$, truncation has no effect on any deletion query. A coloop has zero joint-response row because its one-unit contribution is additive. For each non-coloop u, remove it from the stored matching. If it was matched, at most one augmenting-path repair is required; begin the search from all remaining unmatched peers. Compute the new coloop indicators after removing u. Their difference from the original indicators gives row u of J, with its diagonal set to zero.

The last statement follows directly by comparing the loss of deleting w before and after removing u in Eq. (9). There are at most t repairs and reachability traversals. Including the output matrix, time is $O(tA+t^2)$ and space is $O(A+t^2)$ in the adjacency-list model. The implementation uses integer bitsets for resource membership, whose decoding cost also depends on the representation. The profile uses the same full rank and coloop indicators; its coverage gap comes from the number of distinct touched resources. All these structural quantities are integer computations before normalization.

Global preprocessing computes the recursive levels and the support relation matrices. Its cost depends on the incidence pattern; the stored node relation matrix can be dense. Candidate caches contain the normalized structural features used by the neural readouts.

C Implementation details

The input MLP has widths $6 \to 64 \to 64$, with ReLU after both linear maps and dropout after the first. Each of the two encoder layers transforms $64 \to 64$ with ReLU, aggregates with P_{local}, and applies an update MLP $128 \to 64 \to 64$ with ReLU and dropout between the linear maps. The raw skip is a bias-free $6 \to 64$ linear map. The base scoring MLP has widths $259 \to 64 \to 32 \to 1$, with ReLU after the first two maps and dropout after the first. Candidate variances are clipped below at zero to remove floating-point cancellation artifacts.

The coverage token map has input width $131 = 2 \cdot 64 + 3$, output width 64, bias, and ReLU. Its mean/max head is $128 \to 1$ with bias. The capacity-degree token map is $67 \to 64$ without bias followed by ReLU, and its $128 \to 1$ head has no bias. These contribute 8,577 and 4,416 parameters, respectively. The profile token and output maps have the same dimensions and bias choices as the degree maps, adding 4,416 parameters. The output maps of the coverage, degree, and profile branches are initialized to zero. The full model has 74,306 parameters. Removing either the degree or profile branch leaves 69,890 trainable parameters. Structural quantities are constants during backpropagation.

The implemented base input is $[\mu_S, \sigma_S^2, g_S, c_S, \log(1+s), g_S, c_S]$. Summing the first-layer weights of the duplicate relation blocks gives the effective MLP in Eq. (7). The full model and its matched replacements share this parameterization. All main models and baselines use 600 optimizer updates; the full ablation reference and both component removals use 300 updates. Training uses the full allocated budget. Validation is evaluated at update 1, every two updates, and the last update. A checkpoint replaces the current best only when validation AP improves by more than 10^{-7}. The retained checkpoint is used for test evaluation.

Candidates are packed into chunks of 64 with sparse mean-pooling and relation matrices. Their structural caches are keyed by the support context and candidate. Loss contributions from all chunks and both target folds are accumulated before one optimizer step, with gradient clipping at norm 5. The objective is binary cross-entropy with logits, averaged over all training candidates across both target folds.

D Dataset processing and split details

Table 4: Processed dataset statistics from saved data audits. Positive splits are Train/Val/Test counts. Every split has the same number of negatives as positives. Nodes denote the retained known-ID universe.

Dataset	Nodes	Edges	Mean size	Max	Positive splits
HouseCommittees	1,290	335	35.25	81	201/67/67
SenateBills	294	21,721	9.90	99	13,033/4,344/4,344
EmailEnron	148	1,457	3.09	18	874/291/292
ContactPrimarySchool	242	12,704	2.42	5	7,622/2,541/2,541
ContactHighSchool	327	7,818	2.33	5	4,691/1,564/1,563
NDCSubstances	5,556	6,264	7.96	25	3,758/1,253/1,253
NDCClasses	1,161	1,047	6.11	24	628/209/210

The local files use the names `house-committees`, `senate-bills`, `email-Enron`, `contact-primary-school`, `contact-high-school`, `NDC-substances`, and `NDC-classes`. They are distributed in the collection cited in the main text. The committee data accompany the hypergraph clustering study of Chodrow et al. (2021). The contact datasets derive group events from wearable-sensor measurements in a primary school (Stehlé et al., 2011) and a high school (Mastrandrea et al., 2015), using the maximal-clique construction described by Benson et al. (2018). Preprocessing removes duplicate members inside an event before removing singleton events and collapsing repeated sets. Node IDs from the source remain in the universe even if they no longer occur in a nonsingleton event. For example, NDCSubstances retains 5,556 known IDs, of which 3,438 occur in the processed nonsingleton hyperedges. This convention explains why node counts need not equal the number of active nodes.

The positive split uses a NumPy random generator with seed 42. Counts are rounded by the implementation, as shown in Table 4. Negative sampling uses seed $42 + 2026 = 2068$, with weights based on the complete training-positive graph. A candidate contains distinct members, and rejection excludes all known positive sets and previously sampled negative sets. The code includes a uniform fallback, but the audits record zero fallbacks on all seven datasets. Only the exclusion filter accesses positives outside training; support features and operators use the appropriate training context.

Within training, the shuffled positive index list is alternated into two folds. Target indices select both the positive and its size-matched negative. The opposite positive fold constructs the support hypergraph. Both validation and test candidates are evaluated against both contexts; each candidate's two logits are averaged. Labels are used in the loss or metric calculation, not in candidate structural queries. Isolated members are retained and have zero ordinary support rows, while the neural input still records their standardized structural coordinates.

E Baseline adaptations and budgets

All eight baselines use three input coordinates: standardized log degree, square-root degree, and the nonzero-degree indicator, computed within each support context. The first six share a scoring head that concatenates the mean candidate embedding and $\log(1 + |S|)$, then applies widths $65 \to 64 \to 32 \to 1$. Their encoders are as follows.

- MLP uses no support propagation and applies nodewise feature transformations.
- HGNN uses $D_V^{-1/2} M D_E^{-1} M^\top D_V^{-1/2}$ for normalized incidence propagation (Feng et al., 2019).
- HNHN applies separate nonlinear node-to-hyperedge and hyperedge-to-node updates, with edge-size exponent -1.5 and node-degree exponent -0.5 (Dong et al., 2020).
- AllDeepSets uses learnable multiset maps in both incidence directions, mean aggregation, residual connections, and layer normalization (Chien et al., 2022).
- UniGCNII adds singleton self-loop hyperedges and uses initial residual coefficient 0.1 and identity-mapping coefficient $\log(1 + 0.5/\ell)$ in layer ℓ (Huang & Yang, 2021), following the initial-residual and identity-mapping design of GCNII (Chen et al., 2020).
- FastHyperGCN builds one fixed mediator graph from the initial degree features. Each hyperedge selects endpoints with minimum and maximum degree, adds their pair and mediator connections, and weights each contribution by $1/(2|e| - 3)$. Duplicate pair weights are summed, self-loops are added, and symmetric normalization is applied (Yadati et al., 2019). Endpoint ties use node IDs.

These six implementations adapt the published encoder equations to the shared candidate head. All use hidden width 64 and two encoder layers under the optimization protocol in the main text.

HyperNoRA. Our implementation follows the node-relation scoring and contrastive equations of Li et al. (2026b). Two incidence layers use mean aggregation in both directions, PReLU, and node dropout 0.3. The structural relation graph joins nodes with positive support co-occurrence. For each node, the structural channel averages the embeddings of up to 16 closest nodes by shortest-path distance. The candidate's structural and embedding channels use shared attention maps, a vector-valued sigmoid gate, and a linear classifier. Two augmented support views independently mask

incidences and input features with probability 0.2. The contrastive term pairs each node across views, using up to four structurally closest nodes as negatives and temperature 0.5. The supervised head and projection maps are trained jointly.

S3Hyper. Our implementation follows the substructure-aware readout and self-supervision of Li et al. (2026a). Its two incidence layers use sum aggregation in both directions and the same width, activation, and node dropout as above. A candidate's context includes support hyperedges contained entirely within it. Each member receives the mean of its incident subhyperedge embeddings, with zero for an empty context. The readout fuses this context with member embeddings, applies four-head self-attention with a residual and layer normalization, and uses max pooling followed by a linear classifier.

The auxiliary loss combines node-level contrast, hyperedge-level contrast, and node–hyperedge membership discrimination with equal internal weights. Each view drops nodes from the incidence structure with probability 0.2 and masks features with probability 0.2. Node and edge contrast use all cross-view instances as the reference set, evaluated in blocks of 256. Membership negatives exclude true incidences. We average the membership loss over its four directions and valid incidence pairs. Adaptive node and edge temperatures start at 0.5, have floor 0.1, and use update rate 0.001 and $\rho = 1$. Each support context maintains its own temperatures.

For both methods, the total objective is the mean candidate binary cross-entropy plus 0.1 times the auxiliary loss averaged over the two support contexts. The implementations use the published model equations with the input features, support split, and optimization settings specified here. The full-model comparison therefore includes each predictor's own readout and auxiliary objective.

Numerical implementation. Gradient norms are accumulated in double precision when a single-precision reduction overflows despite finite entries. For S3Hyper, nonfinite-gradient updates use an algebraically identical double-precision readout. The applicable prediction precision is retained for evaluation and recorded with each run. These operations preserve the model equations and loss weights.

Table 5: Trainable parameter counts for the main predictors. Structural preprocessing is outside the learned parameter count.

Model	Parameters
CapacityDegree	74,306
MLP	10,753
HGNN	10,753
HNHN	19,073
AllDeepSets	73,409
UniGCNII	14,785
FastHyperGCN	10,753
HyperNoRA	41,604
S3Hyper	62,663

The experiments use an NVIDIA GeForce RTX 2060, Python 3.10.20, and PyTorch 2.5.1 with CUDA 11.8.

F Additional results

All metrics below use percent units and validation-selected checkpoints. Sample standard deviations (denominator $2 - 1$) measure variation across two initializations on the fixed data split.

Table 6: Test AUC (%), with the same aggregation as Table 1.

Model	House	Senate	Enron	Primary	High	NDC-S	NDC-C	Mean
CapacityDegree	**98.998**	**97.391**	**92.114**	**89.665**	**90.794**	**98.198**	**99.823**	**95.283**
MLP	84.562	73.481	64.926	62.756	57.955	79.491	95.283	74.065
HGNN	96.759	88.963	81.459	72.829	72.718	90.045	97.232	85.715
HNHN	97.137	89.174	79.805	69.165	71.234	95.065	96.958	85.506
AllDeepSets	84.952	72.534	64.965	62.571	58.428	94.349	96.912	76.387
UniGCNII	94.308	86.663	78.827	71.231	71.185	93.370	96.989	84.653
FastHyperGCN	88.060	88.559	82.971	76.734	78.735	89.039	96.873	85.853
HyperNoRA	67.198	71.557	67.198	65.729	64.665	78.749	81.602	70.957
S3Hyper	70.918	94.834	87.786	80.571	81.762	90.221	98.653	86.392

Table 7: Test AP mean $\pm$ sample standard deviation (%), first four datasets.

Model	House	Senate	Enron	Primary
CapacityDegree	99.260 ± 0.084	97.305 ± 0.141	92.137 ± 0.001	91.062 ± 0.492
MLP	82.951 ± 1.290	75.329 ± 0.001	65.598 ± 0.179	63.703 ± 0.008
HGNN	97.124 ± 0.006	88.453 ± 0.549	80.534 ± 1.640	71.952 ± 1.651
HNHN	97.531 ± 0.010	88.835 ± 0.031	79.725 ± 1.014	68.445 ± 0.819
AllDeepSets	83.715 ± 2.985	74.699 ± 0.134	65.023 ± 0.262	63.632 ± 0.009
UniGCNII	94.060 ± 0.097	86.516 ± 0.166	78.332 ± 0.208	70.912 ± 1.102
FastHyperGCN	85.822 ± 0.834	87.896 ± 0.309	82.422 ± 0.852	75.501 ± 0.772
HyperNoRA	73.242 ± 0.244	68.372 ± 0.349	68.829 ± 0.057	67.057 ± 0.160
S3Hyper	71.678 ± 1.318	94.070 ± 0.940	86.987 ± 0.438	80.784 ± 1.352

Table 8: Test AP mean $\pm$ sample standard deviation (%), remaining datasets.

Model	High	NDC-S	NDC-C
CapacityDegree	91.467 ± 0.275	98.386 ± 0.009	99.787 ± 0.010
MLP	58.032 ± 0.180	82.280 ± 0.183	95.674 ± 0.074
HGNN	70.614 ± 0.494	91.197 ± 0.314	97.259 ± 0.077
HNHN	70.685 ± 0.328	95.454 ± 0.302	97.046 ± 0.032
AllDeepSets	57.248 ± 0.553	94.365 ± 0.063	96.863 ± 0.158
UniGCNII	69.366 ± 0.561	93.830 ± 0.096	97.247 ± 0.120
FastHyperGCN	77.151 ± 0.020	90.357 ± 0.254	97.151 ± 0.052
HyperNoRA	67.930 ± 0.432	77.740 ± 0.362	84.575 ± 0.947
S3Hyper	81.917 ± 0.211	90.628 ± 0.811	98.493 ± 0.095

Table 9: Test AUC mean $\pm$ sample standard deviation (%), first four datasets.

Model	House	Senate	Enron	Primary
CapacityDegree	98.998 ± 0.189	97.391 ± 0.139	92.114 ± 0.053	89.665 ± 0.551
MLP	84.562 ± 1.071	73.481 ± 0.046	64.926 ± 0.461	62.756 ± 0.048
HGNN	96.759 ± 0.079	88.963 ± 0.463	81.459 ± 0.964	72.829 ± 1.208
HNHN	97.137 ± 0.047	89.174 ± 0.053	79.805 ± 0.824	69.165 ± 1.124
AllDeepSets	84.952 ± 1.780	72.534 ± 0.009	64.965 ± 0.246	62.571 ± 0.015
UniGCNII	94.308 ± 0.362	86.663 ± 0.039	78.827 ± 0.194	71.231 ± 1.276
FastHyperGCN	88.060 ± 0.977	88.559 ± 0.235	82.971 ± 0.707	76.734 ± 0.762
HyperNoRA	67.198 ± 0.394	71.557 ± 0.075	67.198 ± 0.237	65.729 ± 0.187
S3Hyper	70.918 ± 2.379	94.834 ± 0.673	87.786 ± 0.663	80.571 ± 1.326

Table 10: Test AUC mean ± sample standard deviation (%), remaining datasets.

Model	High	NDC-S	NDC-C
CapacityDegree	90.794 ± 0.330	98.198 ± 0.016	99.823 ± 0.006
MLP	57.955 ± 0.008	79.491 ± 0.099	95.283 ± 0.000
HGNN	72.718 ± 0.180	90.045 ± 0.448	97.232 ± 0.056
HNHN	71.234 ± 0.314	95.065 ± 0.382	96.958 ± 0.005
AllDeepSets	58.428 ± 0.276	94.349 ± 0.148	96.912 ± 0.077
UniGCNII	71.185 ± 0.711	93.370 ± 0.116	96.989 ± 0.186
FastHyperGCN	78.735 ± 0.089	89.039 ± 0.376	96.873 ± 0.064
HyperNoRA	64.665 ± 0.598	78.749 ± 0.341	81.602 ± 0.909
S3Hyper	81.762 ± 0.365	90.221 ± 0.843	98.653 ± 0.141

F.1 Component comparison details

The ablation suite comprises HouseCommittees, EmailEnron, ContactHighSchool, NDCSubstances, and NDCClasses, using the same fixed splits, negatives, support graphs, and initialization seeds as the main comparison. SenateBills and ContactPrimarySchool are excluded from this suite for computational cost. Every included dataset contributes equal weight. Each removal is compared with a full reference trained under the same ablation budget. The no-degree variant retains capacity levels and the profile; the no-profile variant retains capacity levels and the degree branch. Table 2 reports their component-level macro scores.

F.2 Additional mechanism replacements

All comparisons here use HouseCommittees and SenateBills, initialization seeds 42 and 43, and the main training budget. The ordinary-relation reference is the full model without its profile. The coverage-degree replacement changes that reference's degree inputs to coverage responses. The count replacement also replaces capacity-level node features by count levels and computes the degree response from coverage. Its root budgets and eligibility thresholds use C throughout the candidate queries. The parameter-matched coverage profile retains the exact degree branch and substitutes capped coverage for matching rank in Eq. (14): its first coordinate is $c_v(T_v)/D_v$, its second is zero, and its third uses single-deletion losses of c_q. The parameter-matched count profile uses the count replacement and this coverage profile. These structural replacements examine the source of the inputs; the component removals in Table 2 examine the contribution of an entire readout. The original CapacityDegree is an internal reference with a different relation readout and parameter count, evaluated here as a complete predictor.

Table 11: All mechanism references and replacements: equal-weight two-dataset test means.

Model	Parameters	AP (%)	AUC (%)
CapacityDegree	74,306	98.282 ± 0.029	98.194 ± 0.025
Coverage profile	74,306	98.184 ± 0.058	97.998 ± 0.140
Count + coverage profile	74,306	98.249 ± 0.159	98.160 ± 0.284
Ordinary-relation reference	69,890	98.125 ± 0.056	98.071 ± 0.142
Coverage-degree replacement	69,890	98.322 ± 0.004	98.330 ± 0.018
Count replacement	69,890	98.308 ± 0.010	98.340 ± 0.030
Original CapacityDegree	65,730	97.536 ± 0.070	97.561 ± 0.120

Adding the profile to the ordinary-relation reference improves macro AP by 0.157 percentage points and AUC by 0.123 points. The coverage-degree and count replacements score slightly above the full model: their AP advantages are 0.040 and 0.025 points, and their AUC advantages are 0.136 and 0.146 points. Relative to the original internal predictor, the full model gains 0.747 AP and 0.633 AUC points.

F.3 Complete test results: visual summary

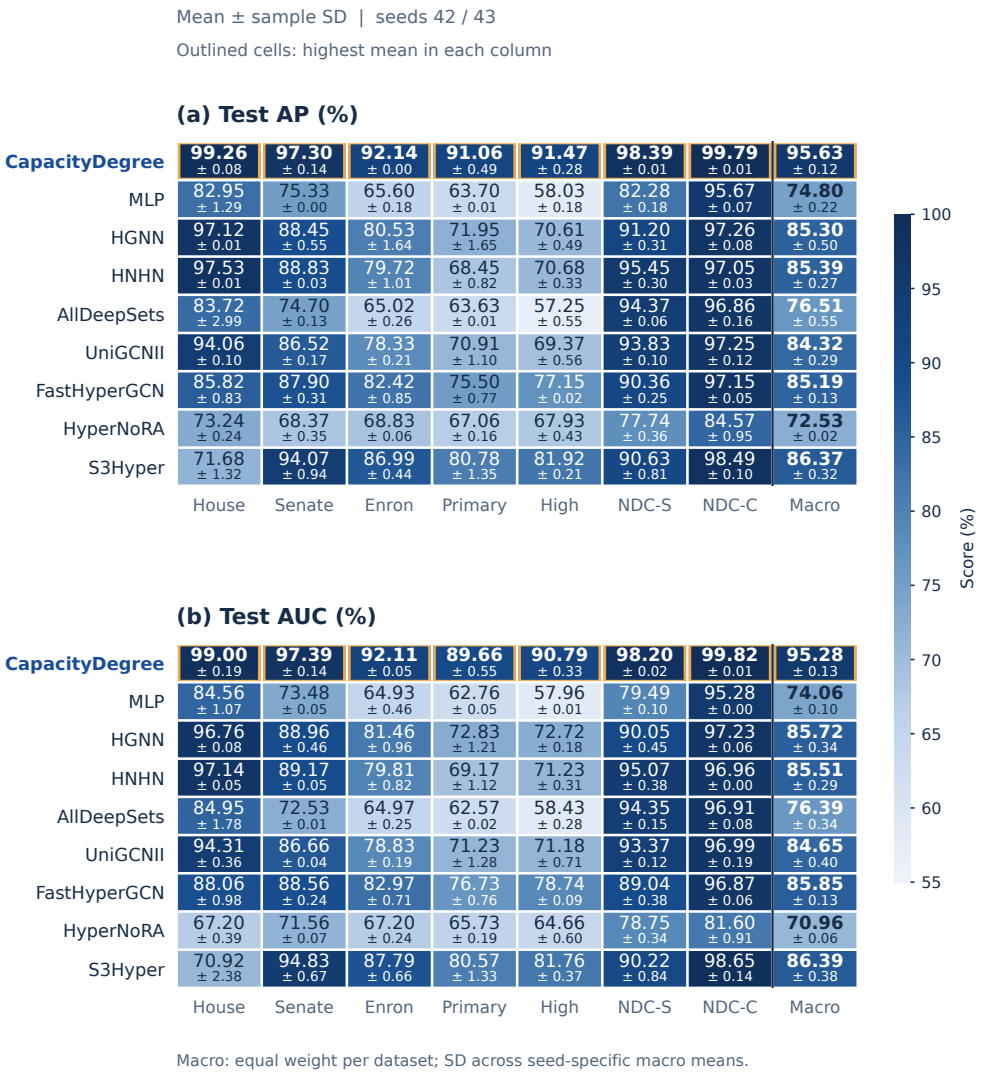


Figure 4: Test AP and AUC for CapacityDegree and all eight baselines. Each cell gives the two-seed mean and sample standard deviation in percent. Macro gives the mean over seven datasets; its standard deviation is computed from the two seed-level macro averages. Outlines mark the highest mean in each column. House/Senate/Enron abbreviate HouseCommittees/SenateBills/EmailEnron, Primary/High are the two contact datasets, and NDC-S/C denote substances/classes.

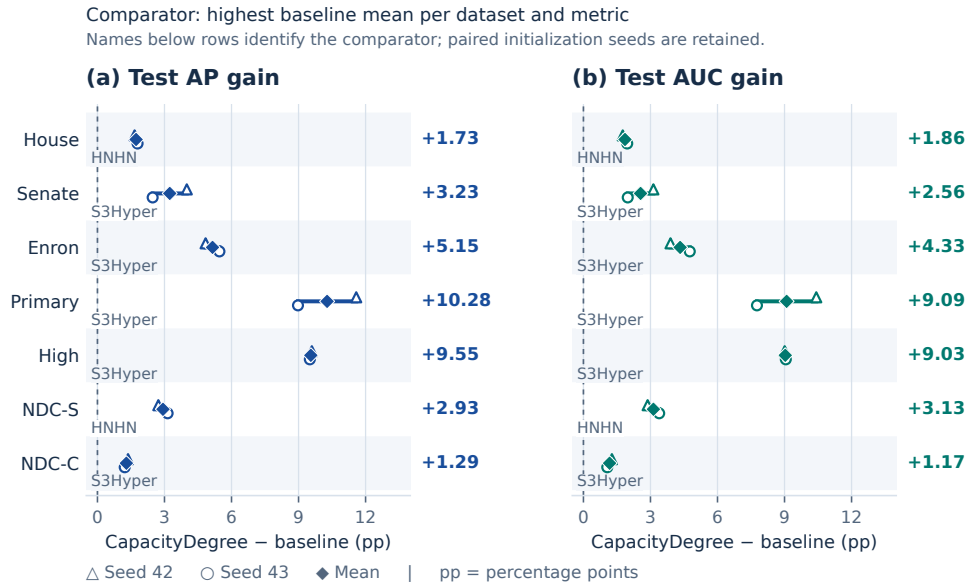


Figure 5: Test gains over the highest-scoring baseline for each dataset and metric, selected by the two-seed mean. Each point compares the same initialization seed; the diamond is the mean difference.

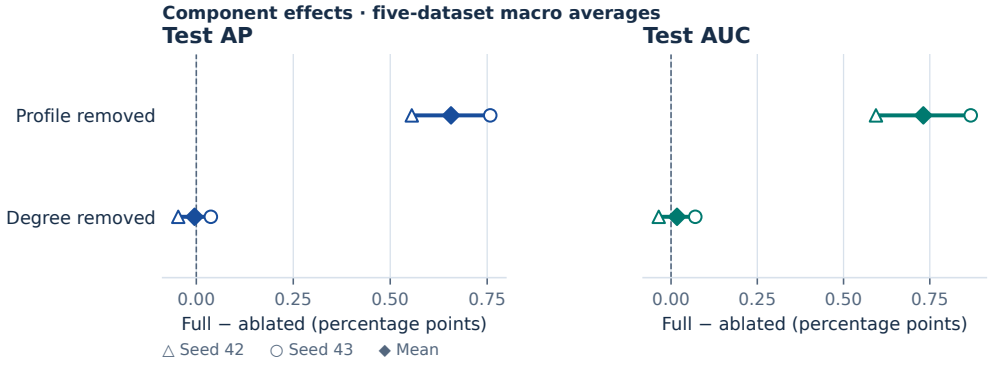


Figure 6: Component ablations aggregated over the five datasets in Appendix F.1. Positive values indicate a higher score for the full model. Each seed's difference is averaged equally over datasets before computing the two-seed mean.