ONE COUPLING, MANY ROLES: READOUT-DEPENDENT ACCESSIBILITY IN GNN REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural network representations with nearly identical supervised linear accessibility can differ sharply in single-prototype recovery, and even a single frozen representation can remain highly linearly accessible while systematically failing single-prototype recovery. On GIN-Cora, we localize this deficit to a single-prototype capacity mismatch within the class-centroid span. This prototype-sensitive failure is accompanied by a recurring degree-conditioned organization on Cora and Amazon-Computers. Training-time manipulation of target-degree message scaling causally shifts the accessibility profile: attenuation in GIN improves prototype accessibility but has different supervised consequences across datasets, whereas amplification in GraphSAGE can induce a large prototype–linear dissociation while largely preserving linear accessibility. These results show that target-degree scaling is a causal lever on representation accessibility, while accessibility itself is readout-dependent rather than a single scalar property of the representation.

1 INTRODUCTION

Similar supervised linear accessibility need not imply similar accessibility under other readout constraints. On Cora (Sen et al., 2008), GIN and GraphSAGE have closely matched supervised linear Macro-F1 point estimates (0.851 vs. 0.845), yet their unsupervised single-prototype recovery differs sharply (0.386 vs. 0.843). When TRAIN labels are used only to place one centroid per class, GIN rises to 0.759. The GIN representation therefore retains class structure that is readily accessible to simple supervised readouts but is difficult to recover with an unsupervised one-prototype-per-class model (Figure 1(a)).

We call this phenomenon *readout-dependent accessibility*: the accessibility exposed by a frozen representation depends on the constraint imposed by the readout, rather than being summarized by a single scalar independent of that constraint. This framing is related to supervised probing of frozen representations and to methodological work emphasizing that probing conclusions depend on the probe and inferential setup (Alain & Bengio, 2016; Belinkov, 2022), as well as to prototype/centroid views of class geometry (Snell et al., 2017; Papyan et al., 2020). Figure 1(b) previews the three readouts used to probe this distinction.

Our contributions are threefold.

First, across five data splits and three model seeds per split, we establish a stable instance of readout-dependent accessibility rather than a single-run anomaly: the same frozen GNN representation can remain highly accessible to supervised linear readout while systematically failing unsupervised single-prototype recovery (Section 2).

Second, we localize the GIN-Cora deficit to a single-prototype capacity mismatch within the class-centroid span U_C, supported by centroid-span preservation, selective multi-prototype rescue, and supervised–unsupervised tessellation mismatch. This prototype-sensitive failure is accompanied by a recurring degree-conditioned organization on Cora and Amazon-Computers (Sections 3–4).

Third, we show that training-time target-degree message scaling can causally shift the learned accessibility profile. GIN attenuation improves prototype accessibility on both Cora and Amazon-

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.

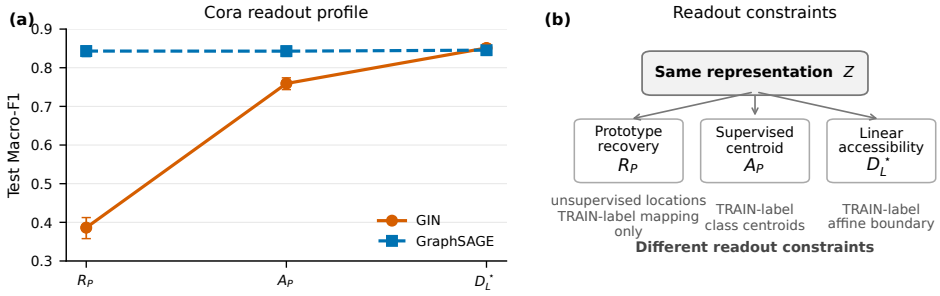


Figure 1: **Same linear accessibility, different prototype recovery. (a)** On Cora, GIN and GraphSAGE have closely matched $D_L^\star$ point estimates but sharply different R_P; supervised centroid placement recovers much of the GIN gap. **(b)** R_P discovers prototypes before TRAIN-label mapping, A_P uses TRAIN labels to place one centroid per class, and $D_L^\star$ learns a validation-selected affine boundary. Error bars denote 95% hierarchical-bootstrap CIs.

Table 1: Three readouts of class structure from a fixed representation.

Readout	Decision geometry	TRAIN-label role	Constraint
R_P	K-Means prototypes	mapping only	one discovered prototype/class
A_P	class centroids	prototype placement	one supervised prototype/class
$D_L^\star$	linear probe	boundary learning	affine boundary

Computers but produces different supervised responses, whereas GraphSAGE amplification can induce a large prototype–linear dissociation while largely preserving linear accessibility. Target-degree scaling is therefore a causal lever on accessibility, but not a scalar "quality knob" with a fixed response across datasets or operator families(Section 5).

2 Readout-Dependent Accessibility in Frozen Representations

We study class accessibility from a fixed representation under different readout constraints. For each trained encoder, the representation is frozen, and all readouts are evaluated by the same TEST Macro-F1. Differences among them therefore reflect how class structure is accessed rather than changes in the representation or evaluation metric.

2.1 Three readout constraints

We compare three readouts that differ in how TRAIN labels may determine the decision geometry (Table 1).

R_P fits $K = C$ K-Means (MacQueen, 1967) prototypes without label-based placement; TRAIN labels are used only afterward to map clusters to classes. A_P instead uses TRAIN labels to place one centroid per class. Because both retain the same Euclidean one-prototype decision form, we define their difference as the *discovery gap*

$$\Delta_{\text{disc}} = A_P - R_P.$$

A positive Δ_{disc} therefore measures the accessibility gain from supervised prototype placement under the same single-prototype decision form. $D_L^\star$ further relaxes the one-centroid constraint by learning a validation-selected affine boundary. Thus, R_P versus A_P isolates supervised prototype placement, whereas A_P versus $D_L^\star$ tests the additional benefit of boundary flexibility. Full definitions and fitting procedures are given in Appendix A.

2.2 Stable accessibility dissociation on Cora

The full profiles underlying the introductory contrast are

$$\text{GIN} : (R_P, A_P, D_L^\star) = (0.386,\ 0.759,\ 0.851), \qquad \text{GraphSAGE} : (0.843,\ 0.843,\ 0.845).$$

GIN therefore exhibits a large discovery gap, $\Delta_{\text{disc}} = 0.373$ (95% hierarchical-bootstrap CI $[0.337, 0.409]$), whereas GraphSAGE is approximately aligned ($\Delta_{\text{disc}} = -0.0002$, 95% CI $[-0.0014, 0.0005]$).

Despite their closely matched linear accessibility in point estimate, the two architectures differ sharply in prototype recovery. The paired discovery-gap contrast has the same direction across all five data splits. The GIN profile therefore represents a stable within-representation dissociation: supervised centroid and linear readouts recover class structure that an unlabeled single-prototype readout systematically misses.

2.3 Evaluation and inferential scope

Unless stated otherwise, results use five data splits and three model seeds per split, with K-Means repetitions aggregated within each representation run. Uncertainty is estimated by split–model-seed hierarchical bootstrap; full procedures are given in Appendix B. Frozen-representation measurements are treated as observational diagnostics, consistent with the distinction between probe-based association and intervention-based causal evidence (Belinkov, 2022); causal claims are restricted to retrained operator interventions.

3 Localizing the Prototype-Recovery Deficit

The GIN-Cora discovery gap admits two explanations. Class-relevant structure may lie outside the span determined by the class centroids, or the deficit may already exist within that span because one prototype per class is too restrictive. We distinguish these possibilities by restricting the frozen representation to the class-centroid span and then varying prototype capacity within it.

3.1 The deficit persists in the class-centroid span

Let

$$U_C = \operatorname{span}\{\mu_1 - \bar{\mu}, \ldots, \mu_C - \bar{\mu}\},$$

where μ_c is the TRAIN centroid of class c and $\bar{\mu}$ is their mean. We project each frozen representation onto U_C and $U_C^{\perp}$ and refit the complete TRAIN-only recovery pipeline in each projected space, for each architecture independently (Appendix C).

For GIN, restricting the representation to U_C leaves prototype recovery essentially unchanged (Figure 2(a)):

$$R_P^{\text{full}} = 0.386, \qquad R_P^{U_C} = 0.375,$$

while supervised centroid accessibility remains much higher($A_P = 0.759$). The discovery deficit therefore does not require centroid-orthogonal directions; it is already present within U_C. Consistently, retaining only $U_C^{\perp}$ sharply reduces supervised linear accessibility ($D_L^{\star} : 0.851 \to 0.122$). This does not imply that $U_C^{\perp}$ contains no class-related information, only that it is unnecessary for the observed prototype-recovery failure.

GraphSAGE provides a complementary control: its U_C-projected prototype, centroid, and linear accessibility are virtually unchanged from the full-space values ($R_P^{U_C} = 0.843$, $A_P^{U_C} = 0.843$, $D_L^{U_C} = 0.846$). The projection therefore does not by itself induce a discovery deficit; the single-prototype failure remains specific to the GIN representation.

3.2 Additional prototypes rescue the deficit within U_C

We next relax only the prototype-capacity constraint while keeping the learned representation fixed. K-Means is fitted with

$$K = qC, \qquad q \in \{1, 2, 3, 4\},$$

using the same TRAIN-only many-to-one cluster–class mapping rule for every q, including $q = 1$. Hence differences across q reflect additional prototype capacity rather than a change in mapping criterion.

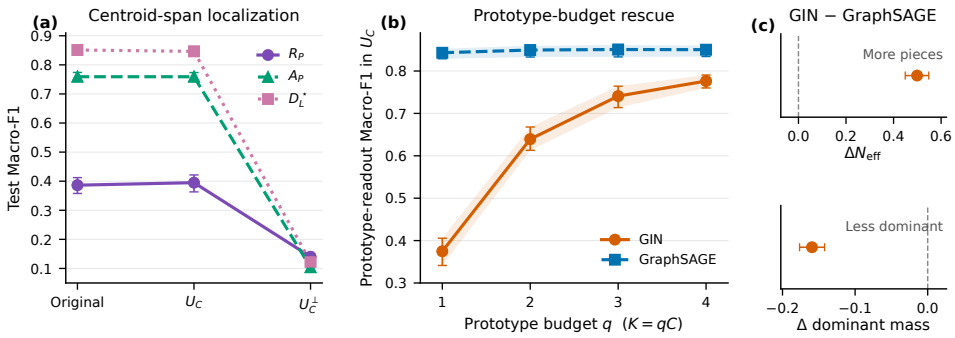


Figure 2: **The GIN-Cora prototype deficit is localized to a single-prototype mismatch within the centroid span.** **(a)** Projection onto U_C preserves low GIN prototype recovery, whereas $U_C^\perp$ retains little supervised linear accessibility. **(b)** Increasing prototype capacity within U_C strongly rescues GIN but barely changes GraphSAGE. **(c)** GIN shows a larger mismatch between supervised centroid and unsupervised K-Means partitions. Error bars denote 95% hierarchical-bootstrap confidence intervals.

Within U_C, GIN recovery rises from approximately 0.375 at $q = 1$ to 0.776 at $q = 4$, whereas GraphSAGE changes only from approximately 0.843 to 0.850 under the same prototype-budget increase (Figure 2(b)). The selective rescue identifies the GIN-Cora deficit operationally as a *single-prototype capacity mismatch within* U_C: class structure remains accessible in the centroid span, but a single discovered Euclidean prototype per class is too restrictive to recover it.

This result does not establish discrete semantic multimodality. Additional prototypes can also accommodate anisotropy, density variation, interleaving, or other departures from a one-prototype geometry.

3.3 The $q = 1$ partition mismatches supervised centroid regions

To test whether the prototype-budget rescue corresponds to an existing partition mismatch, we compare the $K = C$ K-Means partition with the supervised centroid Voronoi partition inside U_C. For each supervised centroid region, N_{eff} measures the effective number of contributing K-Means regions, while M_{dom} measures the mass assigned to its dominant discovered region; complete definitions are given in Appendix C.

GIN exhibits substantially stronger mismatch than GraphSAGE(Figure 2(c)):

$$\Delta N_{\text{eff}} = 0.4990 \quad (95\%\ \text{CI}\ [0.4492, 0.5482]),$$

and

$$\Delta M_{\text{dom}} = -0.1594 \quad (95\%\ \text{CI}\ [-0.1765, -0.1420]).$$

Thus, relative to GraphSAGE, supervised centroid regions in GIN are distributed across more discovered K-Means regions and place less mass in their dominant region. This partition-level result provides an independent geometric counterpart to the prototype-budget rescue.

Together, centroid-span preservation, selective multi-prototype rescue, and tessellation mismatch support a single diagnosis: the GIN-Cora discovery deficit is a single-prototype capacity mismatch already present within U_C. The diagnosis characterizes the form of the failure, without identifying what structural factor organizes it.

4 Recurring Degree-Conditioned Organization

The single-prototype mismatch within U_C leaves open whether the incompatible within-class geometry is systematically aligned with graph structure. We examine target-node degree as a candidate structural coordinate and test whether prototype-sensitive representations exhibit recurring degree-conditioned organization. These analyses are observational; causal claims are reserved for the training-time interventions in Section 5.

4.1 DEGREE-CONDITIONED ORGANIZATION ON CORA

Target degree is a natural coordinate because native GIN sum aggregation retains an explicit dependence on neighborhood cardinality, whereas mean aggregation removes this dependence at the pre-transform neighborhood-summary level (Xu et al., 2018). Degree-sensitive behavior has also been documented in node prediction and representation learning (Wang et al., 2022; Ju et al., 2023; Subramonian et al., 2024). To separate degree from class identity, we construct TRAIN-only degree strata within each class and freeze them for TEST. We measure partition-level dependence with $I_{\mathrm{adj}}(K; D_{\mathrm{bin}} \mid Y)$ and direct within-class geometry with G_{degree} in U_C; full definitions are given in Appendix D.

Both diagnostics are substantially larger for GIN than GraphSAGE on Cora:

$$\Delta I_{\mathrm{adj}} = 0.1071 \ (95\% \text{ CI } [0.0784, 0.1371]), \qquad \Delta G_{\mathrm{degree}} = 0.2449 \ (95\% \text{ CI } [0.2122, 0.2802]),$$

with the GIN–GraphSAGE direction preserved across all five data splits. A supporting four-architecture Cora characterization places GIN highest on fixed-dose prototype sensitivity and both degree-conditioned diagnostics, with GCN intermediate and GATv2/GraphSAGE substantially smaller (Appendix D). This provides supporting breadth for the structural signature without treating the panel as an architecture-level causal ranking.

4.2 PHENOTYPE-CONDITIONED RECURRENCE ON AMAZON-COMPUTERS

Amazon-Computers (Shchur et al., 2018) was selected from a previously frozen Phase-I analysis because it already exhibited a discovery-dominant phenotype, before the prototype-budget and degree-conditioned measurements reported here were inspected. We therefore treat it as phenotype-conditioned recurrence rather than blind replication.

The same pattern recurs (Figure 4). Additional prototype capacity yields a larger recovery gain for GIN than GraphSAGE (0.3535 versus 0.2071), and the paired GIN–GraphSAGE structural contrasts are

$$\Delta I_{\mathrm{adj}} = 0.2166 \ (95\% \text{ CI } [0.1845, 0.2468]), \qquad \Delta G_{\mathrm{degree}} = 0.2028 \ (95\% \text{ CI } [0.1515, 0.2565]),$$

again with the same direction across all five data splits. Across both datasets, prototype sensitivity therefore co-occurs with stronger degree-conditioned organization, identifying target-degree dependence as a concrete operator-level factor for controlled intervention. Full recurrence diagnostics are reported in Appendix D.

5 INTERVENING ON TARGET-DEGREE MESSAGE SCALING

The localization and recurrence analyses are observational. We therefore manipulate target-degree message scaling during training—attenuating it in GIN and amplifying it in GraphSAGE—and test whether the resulting accessibility profile changes (Figure 3).

5.1 OPPOSITE-DIRECTION TARGET-DEGREE INTERVENTIONS

Let

$$\bar{h}^{(\ell)}_{\mathcal{N}(v)} = \mathrm{Mean}_{u \in \mathcal{N}(v)} \, h^{(\ell)}_u$$

denote the mean neighborhood representation of target node v, with $\widetilde{d}_v = \max(d_v, 1)$. Native GIN (Xu et al., 2018) uses a neighborhood sum, which can be written as $d_v \, \bar{h}^{(\ell)}_{\mathcal{N}(v)}$, corresponding to an explicit target-degree prefactor d_v^1. Native mean-aggregating GraphSAGE (Hamilton et al., 2017) uses $\bar{h}^{(\ell)}_{\mathcal{N}(v)}$ alone, corresponding to d_v^0. We manipulate this target-degree dependence in opposite directions while retaining the remaining architecture-specific computation.

For GIN, the neighborhood contribution is attenuated as

$$m_v^{\mathrm{GIN}}(\alpha) = \widetilde{d}_v^{-\alpha} \sum_{u \in \mathcal{N}(v)} h^{(\ell)}_u,$$

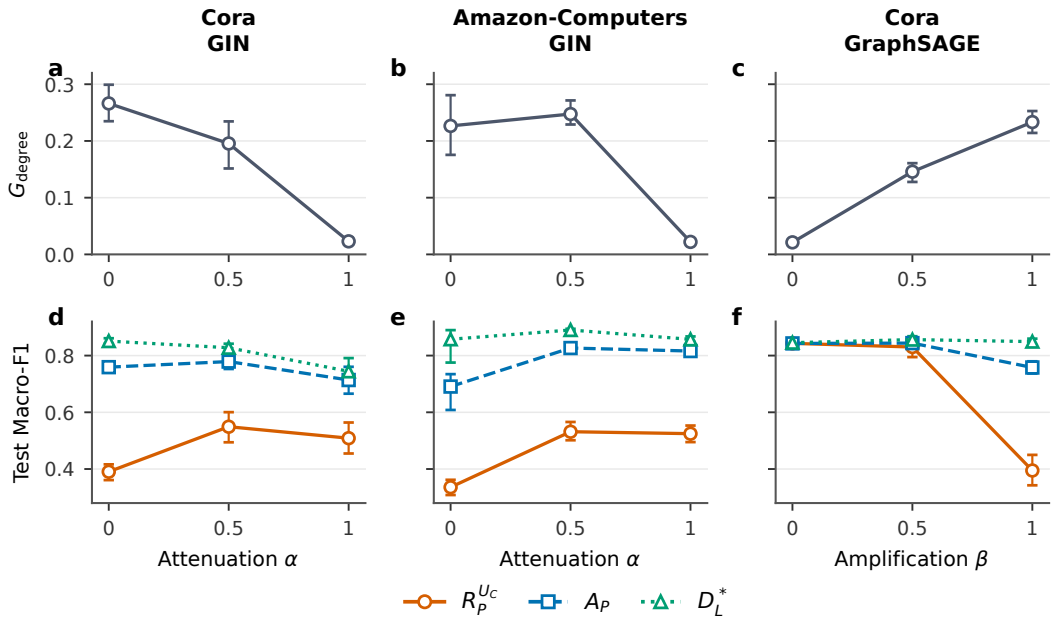


Figure 3: **Target-degree interventions produce distinct accessibility responses.** Cora–GIN attenuation (**a,d**), Amazon-Computers–GIN attenuation (**b,e**), and Cora–GraphSAGE amplification (**c,f**). Top: G_{degree}; bottom: $R_P^{U_C}$, A_P, and $D_L^\star$. hierarchical-bootstrap CIs.

giving an effective target-degree exponent $1 - \alpha$. This factor modifies only the neighborhood sum; the self term and MLP placement are unchanged from native GIN. Thus, $\alpha = 0$ recovers native GIN, whereas $\alpha = 1$ yields a mean-like neighborhood summary. Cora uses $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and Amazon-Computers uses $\alpha \in \{0, 0.5, 1\}$.

For GraphSAGE, the native mean neighborhood representation is amplified as

$$m_v^{\text{SAGE}}(\beta) = \widetilde{d}_v^{\beta} \bar{h}_{\mathcal{N}(v)}^{(\ell)}, \qquad \beta \in \{0, 0.5, 1\}.$$

The factor is applied after mean aggregation and before the learned neighborhood transform. It modifies only the neighborhood branch; the root branch is unchanged. Thus, $\beta = 0$ recovers native GraphSAGE, whereas increasing β introduces stronger explicit target-degree dependence into the neighborhood branch.

The two intervention families therefore traverse the same pre-transform target-degree exponent range in opposite directions. This correspondence does not make the architectures equivalent: $\alpha = 1$ is not GraphSAGE, $\beta = 1$ is not GIN, and α and β are not quantitatively matched treatment doses.

Each treatment retrains the encoder under a paired design over five data splits and three model seeds; within each split–seed cell, all treatment values share the same initialization and data order. The resulting effects therefore measure the causal consequence of changing target-degree message scaling, including the model's adaptation to that change, rather than a post-hoc transformation of a fixed embedding. Full operator definitions and intervention-integrity checks are provided in Appendix E.

5.2 Attenuation improves prototype access, but its supervised consequences differ

We attenuate the target-degree exponent in GIN, where the native representation exhibits a large discovery gap. Figure 3(a,d) shows the full dose trajectory. On Cora, the intervention successfully suppresses the targeted structural organization: G_{degree} decreases from 0.266 at $\alpha = 0$ to 0.023 at $\alpha = 1$. Prototype accessibility inside the centroid span improves, but not monotonically: $R_P^{U_C}$ rises from 0.390 to 0.549 at $\alpha = 0.5$ and remains elevated at 0.509 at $\alpha = 1$. The largest observed recovery therefore occurs at moderate rather than maximal attenuation. We preregistered a

stronger prediction: that this recovery would occur without degrading supervised accessibility. That clean-repair prediction is rejected: at $\alpha = 1$, A_P decreases by 0.045 and $D_L^\star$ by 0.107 relative to native GIN. Hence, on Cora, stronger attenuation improves single-prototype recovery but is not a semantically free repair.

The same attenuation family produces a different supervised response on Amazon-Computers. Figure 3(b,e) shows the corresponding trajectories. Increasing α raises $R_P^{U_C}$ from 0.3358 to 0.5316 at $\alpha = 0.5$ and 0.5247 at $\alpha = 1$. Unlike Cora, supervised centroid accessibility improves: A_P increases from 0.6908 to 0.8266 and 0.8160, respectively. $D_L^\star$ follows the trajectory $(0.8575, 0.8902, 0.8573)$ across $\alpha = 0, 0.5, 1$. Its endpoint contrast is negligible ($\Delta D_L^\star = -0.0002$, 95% CI $[-0.0343, 0.0807]$), although the preregistered non-inferiority criterion is not established. At the endpoint, G_{degree} is strongly suppressed, decreasing from 0.2265 to 0.0220.

The cross-dataset comparison separates two conclusions: prototype accessibility improves in both GIN settings, but the supervised response does not transport—Cora shows prototype gain with centroid and linear cost, whereas Amazon shows joint prototype–centroid improvement with linear accessibility preserved. Target-degree attenuation is therefore a causal lever on accessibility, not a uniform "quality improvement" operation.

5.3 Amplification induces a discovery-side dissociation in GraphSAGE

Native GraphSAGE on Cora is initially aligned across readouts, with $R_P^{U_C} \approx A_P \approx D_L^\star \approx 0.84$ at $\beta = 0$. Amplifying target-degree dependence tests whether this aligned profile can be driven toward a discovery-side dissociation.

Increasing the target-degree exponent produces a large and structured shift. Figure 3(c,f) shows the corresponding trajectories. At $\beta = 1$, G_{degree} increases from 0.021 to 0.233, while $R_P^{U_C}$ falls from 0.843 to 0.395. Supervised centroid accessibility also decreases, but substantially less, from 0.843 to 0.758, whereas linear accessibility remains essentially unchanged ($0.845 \to 0.849$). The profile therefore shifts from near alignment to $R_P^{U_C} \ll A_P < D_L^\star$. Together with the GIN attenuation results, this provides opposite-direction training-time intervention evidence: decreasing the target-degree exponent in GIN increases prototype accessibility, whereas increasing it in GraphSAGE can induce a large prototype–linear accessibility gap from an initially aligned representation. The accessibility gap is therefore causally movable in both intervention directions, although the responses are neither symmetric nor architecture-equivalent.

The preregistered strong reciprocal criterion, however, is not fully satisfied. The manipulation and discovery-shift gates pass, but the supervised centroid guardrail fails because A_P decreases by 0.085; the linear-accessibility guardrail remains satisfied. The intervention therefore preserves linear accessibility, not centroid accessibility, and should not be interpreted as a clean reciprocal transformation of native GraphSAGE into GIN.

Increasing the prototype budget clarifies the induced failure further: at $\beta = 1$, raising the budget from $q = 1$ to $q = 4$ improves $R_P^{U_C}$ from 0.395 to 0.787, recovering approximately 87% of the $q = 1$ drop. This recovery is consistent with the prototype-capacity mismatch localized in Section 3, indicating that the intervention induces a similar form of single-prototype incompatibility rather than a generic loss of class information.

5.4 Distinct accessibility responses to target-degree scaling

The three intervention settings produce qualitatively different accessibility responses (Table 2). Cora–GIN attenuation improves prototype accessibility while reducing centroid and linear accessibility; Amazon–GIN attenuation improves both prototype and centroid accessibility with little endpoint change in linear accessibility; and Cora–GraphSAGE amplification sharply reduces prototype accessibility while largely preserving linear accessibility. Target-degree scaling is therefore a causal lever on the learned accessibility profile, but not a scalar "quality knob" that uniformly improves or degrades all readouts.

The intermediate treatments also show that degree-conditioned organization and accessibility do not move in lockstep. In GraphSAGE, at $\beta = 0.5$, G_{degree} has already increased from 0.021 to 0.146, while $R_P^{U_C}$ changes only from 0.843 to 0.830. Conversely, on Amazon-Computers at $\alpha = 0.5$,

Table 2: Endpoint accessibility changes from the native setting to treatment level 1. Brackets give marginal 95% paired hierarchical-bootstrap CIs. The attenuation parameter α and amplification parameter β define distinct interventions.

Setting	$\Delta R_P^{U_C}$	ΔA_P	$\Delta D_L^\star$
Cora–GIN	$+0.118$ $[+0.071, +0.169]$	-0.045 $[-0.105, +0.010]$	-0.107 $[-0.155, -0.057]$
Amazon–GIN	$+0.189$ $[+0.148, +0.233]$	$+0.125$ $[+0.081, +0.206]$	-0.0002 $[-0.034, +0.081]$
Cora–GraphSAGE	-0.448 $[-0.494, -0.400]$	-0.085 $[-0.102, -0.068]$	$+0.004$ $[-0.004, +0.010]$

$R_P^{U_C}$ and A_P improve substantially ($0.336 \to 0.532$ and $0.691 \to 0.827$), even though G_{degree} has not decreased ($0.227 \to 0.247$). These trajectories are inconsistent with a simple scalar-mediation account, but do not identify a mediator. We therefore treat G_{degree} as a post-treatment structural diagnostic rather than an identified causal mediator.

Training-time target-degree scaling can therefore redistribute accessibility across readouts, while the form of that redistribution depends on the dataset and operator family.

6 Related Work

Representation readouts and prototype accessibility. Linear probes are widely used to assess what task-relevant structure is accessible from frozen representations through a low-complexity supervised decoder (Alain & Bengio, 2016). Probe design itself matters: probing conclusions depend on probe capacity, controls, datasets, and inferential scope (Hewitt & Liang, 2019; Belinkov, 2022). Recent work has further examined linear evaluation through risk decomposition (Dubois et al., 2023), comparisons across protocols (Marks et al., 2025), and alternative decoding bottlenecks (Laina et al., 2022). Prototype-based methods provide a complementary geometric constraint: Prototypical Networks summarize each class with one prototype (Snell et al., 2017), while mixture-prototype methods relax the single-center assumption (Allen et al., 2019; Yang et al., 2020). Related work has also characterized clustering structure (Weng et al., 2025), class-mean geometry under Neural Collapse (Papyan et al., 2020; Kothapalli et al., 2023), and nearest-class-center mismatch (Ben-Shaul & Dekel, 2022). Together, these strands provide complementary notions of linear, centroid, and prototype accessibility, but they are typically studied under their own evaluation protocols rather than compared on the same frozen representation.

Degree-sensitive message passing and structural organization. Target degree enters GNN message passing in architecture-dependent ways. GCN uses degree-normalized propagation (Kipf & Welling, 2017), GraphSAGE includes mean aggregation (Hamilton et al., 2017), and GIN uses sum aggregation to preserve multiset information (Xu et al., 2018). PNA makes degree dependence explicit through multiple aggregators and degree scalers (Corso et al., 2020). Degree-sensitive prediction and representation behavior has been studied through degree-fairness and structural-fairness analyses (Liu et al., 2023; Wang et al., 2022), test-time augmentation for low-degree nodes (Ju et al., 2023), and analyses connecting degree bias with graph filters, homophily, and neighborhood diversity (Subramonian et al., 2024). These studies establish several forms of degree-sensitive behavior, but do not ask how target degree is associated with the class-conditional organization of a learned representation.

Operator-induced geometry and training-time intervention. Graph propagation and normalization can systematically reshape learned representations. Oversmoothing analyses connect repeated propagation with feature contraction and restricted graph-dependent subspaces (Li et al., 2018; Oono & Suzuki, 2019). DropEdge changes propagation during training (Rong et al., 2020), while PairNorm and differentiable group normalization explicitly modify normalization or representation geometry to counter oversmoothing (Zhao & Akoglu, 2019; Zhou et al., 2020). GCNII and GraphNorm modify propagation or normalization for deeper or more stable training (Chen et al., 2020; Cai et al., 2021), and GRANOLA makes normalization graph-adaptive (Eliasof et al., 2024). Together, these works show that message-passing and normalization choices can materially change

learned representations and GNN behavior. We study the complementary question of how controlled target-degree scaling redistributes accessibility across prototype, centroid, and linear readouts.

7 Discussion, Limitations, and Future Work

Our results support a readout-dependent view of representation accessibility. A frozen representation can remain highly accessible to supervised linear readout while failing unsupervised single-prototype recovery; the failure is not due to missing class information, but to a mismatch between the learned class-conditional geometry and the one-prototype-per-class constraint. For GIN on Cora, the deficit persists within the class-centroid span and is selectively rescued by additional prototypes. The implication is not that representation quality is arbitrary, but that it cannot be summarized by a single readout-independent scalar.

The degree-conditioned organization provides a recurring structural correlate of this prototype-sensitive failure, but its interpretation is deliberately limited. Recurrence on Cora and Amazon-Computers supports an association between degree-conditioned organization and the accessibility phenotype; it does not by itself establish degree as the cause of that phenotype. Likewise, multi-prototype rescue supports a single-prototype capacity mismatch but does not establish semantic multimodality. The intervention results further separate the structural diagnostic from the accessibility outcome: changing target-degree message scaling during training causally shifts the learned accessibility profile, yet G_{degree} and accessibility do not move in lockstep. We therefore treat G_{degree} as a post-treatment structural diagnostic rather than an identified causal mediator.

The causal evidence is also bounded by the intervention families studied here. GIN attenuation and GraphSAGE amplification manipulate the same pre-transform target-degree scaling dimension in opposite directions, but their endpoints are not architecture-equivalent and α and β are not matched treatment doses. Because each treatment retrains the encoder, the observed effects include both the modified operator and the model's adaptation to it. Across Cora and Amazon-Computers, the resulting redistribution differs by dataset and operator family rather than following a universal tradeoff. A natural next step is to ask which other operator-level factors and graph regimes produce the same accessibility dissociation, and whether such interventions can selectively improve one readout without degrading others.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International conference on machine learning*, pp. 232–241. PMLR, 2019.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Ido Ben-Shaul and Shai Dekel. Nearest class-center simplification through intermediate layers. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pp. 37–47. PMLR, 2022.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks?, 2022.

Tianle Cai, Shengjie Luo, Keyulu Xu, Di He, Tie-yan Liu, and Liwei Wang. Graphnorm: A principled approach to accelerating graph neural network training. In *International Conference on Machine Learning*, pp. 1204–1215. PMLR, 2021.

Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International conference on machine learning*, pp. 1725–1735. PMLR, 2020.

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. *Advances in neural information processing systems*, 33:13260–13271, 2020.

Yann Dubois, Tatsunori Hashimoto, and Percy Liang. Evaluating self-supervised learning via risk decomposition. In *International Conference on Machine Learning*, pp. 8779–8820. PMLR, 2023.

Moshe Eliasof, Beatrice Bevilacqua, Carola-Bibiane Schönlieb, and Haggai Maron. Granola: Adaptive normalization for graph neural networks. *Advances in Neural Information Processing Systems*, 37, 2024.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: Mitigating degree bias for graph neural networks via test-time augmentation. *Advances in Neural Information Processing Systems*, 36, 2023.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. URL `https://mlanthology.org/iclr/2017/kipf2017iclr-semi/`.

Vignesh Kothapalli, Tom Tirer, and Joan Bruna. A neural collapse perspective on feature evolution in graph neural networks. *Advances in Neural Information Processing Systems*, 36:14134–14191, 2023.

Iro Laina, Yuki M Asano, and Andrea Vedaldi. Measuring the interpretability of unsupervised representations via quantized reverse probing. *arXiv preprint arXiv:2209.03268*, 2022.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. On generalized degree fairness in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4525–4533, 2023.

J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.

Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, and Pietro Perona. A closer look at benchmarking self-supervised pre-training with image classification. *International Journal of Computer Vision*, 133(8):5013–5025, 2025.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

Vardan Papyan, Xiao Y Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. In *AI Magazine*, volume 29, pp. 93–106, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan G"unnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS*, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Arjun Subramonian, Jian Kang, and Yizhou Sun. Theoretical and empirical insights into the origins of degree bias in graph neural networks. *Advances in Neural Information Processing Systems*, 37: 8193–8239, 2024.

Ruijia Wang, Xiao Wang, Chuan Shi, and Le Song. Uncovering the structural fairness in graph contrastive learning. *Advances in Neural Information Processing Systems*, 35, 2022.

Xi Weng, Jianing An, Xudong Ma, Binhang Qi, Jie Luo, Xi Yang, Jin Song Dong, and Lei Huang. Clustering properties of self-supervised learning. *arXiv preprint arXiv:2501.18452*, 2025.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *Computer Vision – ECCV 2020*, pp. 763–778. Springer, 2020.

Lingxiao Zhao and Leman Akoglu. Pairnorm: Tackling oversmoothing in gnns. *arXiv preprint arXiv:1909.12223*, 2019.

Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. Towards deeper graph neural networks with differentiable group normalization. *Advances in Neural Information Processing Systems*, 33, 2020.

A Accessibility Readout Definitions

All three readouts operate on the same frozen representation after a TRAIN-fitted standardization transform. The definitions below specify the complete fitting, label-usage, and TEST-evaluation procedures summarized in Table 1.

A.1 Shared representation preprocessing

Consider a node-classification dataset

$$\mathcal{G} = (V, E, X, Y),$$

where V and E denote nodes and edges, X denotes node features, and $Y \in \{1, \dots, C\}^{|V|}$ denotes class labels. A trained graph encoder f_θ maps node v_i to

$$z_i = f_\theta(X, E)_i,$$

giving the representation matrix

$$Z \in \mathbb{R}^{|V| \times d}.$$

For every trained representation, the encoder is frozen before accessibility measurement. A per-dimension standardization transform is fitted using TRAIN nodes only and then applied unchanged to TRAIN, VAL, and TEST. Prototype, centroid, and linear readouts are all evaluated in this common standardized space.

Analyses of an already-trained representation are therefore observational with respect to the encoder. In the intervention experiments, the operator is modified during training and a new representation is learned under each treatment; the resulting representation is then frozen and evaluated by the same protocol.

A.2 UNSUPERVISED PROTOTYPE RECOVERY R_P

For a dataset with C classes, prototype recovery fits $K = C$ K-Means prototypes using TRAIN embeddings:

$$\{\hat{\mu}_k\}_{k=1}^{C} = \mathrm{KMeans}(Z_{\mathrm{train}}, K = C).$$

Class labels do not determine prototype locations or the induced Voronoi partition. Because K-Means cluster indices have no class semantics, TRAIN labels are used after clustering to construct the cluster–class contingency matrix. Hungarian matching yields a one-to-one mapping

$$\pi : \{1, \ldots, C\} \to \{1, \ldots, C\}.$$

Both the fitted prototype locations and the mapping π are frozen before TEST evaluation. For a TEST representation z,

$$k^*(z) = \arg\min_k \|z - \hat{\mu}_k\|_2^2,$$

and

$$\hat{y}_R(z) = \pi(k^*(z)).$$

Prototype recovery is

$$R_P = F_1^{\mathrm{macro}}\left(Y_{\mathrm{test}}, \hat{Y}_R\right).$$

R_P is therefore not a completely label-free clustering score: TRAIN labels are used to assign class semantics to discovered clusters. Its defining restriction is that labels do not participate in prototype placement. Consequently, R_P measures whether a one-prototype-per-class organization can be discovered from the representation geometry itself.

A.3 SUPERVISED CENTROID ACCESSIBILITY A_P

Supervised centroid accessibility retains the same one-prototype-per-class Euclidean decision form but allows TRAIN labels to determine prototype locations.

For class c, the TRAIN centroid is

$$\mu_c = \frac{1}{|V_c^{\mathrm{train}}|} \sum_{i \in V_c^{\mathrm{train}}} z_i.$$

A TEST representation is classified as

$$\hat{y}_A(z) = \arg\min_c \|z - \mu_c\|_2^2,$$

and

$$A_P = F_1^{\mathrm{macro}}\left(Y_{\mathrm{test}}, \hat{Y}_A\right).$$

The distinction between R_P and A_P is therefore prototype placement:

$$R_P: \quad \text{prototype locations discovered from unlabeled geometry,}$$

whereas

$$A_P: \quad \text{prototype locations specified by TRAIN labels.}$$

We define the discovery gap

$$\Delta_{\mathrm{disc}} = A_P - R_P.$$

A positive discovery gap does not imply that class-related information is absent from the representation. It measures the accessibility gained when TRAIN labels are allowed to specify one prototype per class while retaining the same basic Euclidean decision form.

A.4 SUPERVISED LINEAR ACCESSIBILITY $D_L^\star$

A_P remains constrained to one centroid per class and therefore induces a Euclidean Voronoi partition. We additionally measure accessibility under a more flexible, but still low-complexity, supervised affine readout.

We deliberately restrict this readout to a pre-specified low-complexity linear family, since probe expressivity can itself complicate the interpretation of probing accuracy (Hewitt & Liang, 2019).

The frozen probe family contains Logistic Regression, Linear SVM, and Ridge Classifier. Each candidate is fitted using TRAIN nodes. Probe family and hyperparameters are selected using VAL Macro-F1, and TEST is evaluated only after this selection.

Let $\mathcal{H}_L$ denote the frozen candidate family. The selected probe is

$$g_L^* = \arg\max_{g \in \mathcal{H}_L} F_1^{\text{macro}}\left(Y_{\text{val}}, g(Z_{\text{val}})\right),$$

and

$$D_L^\star = F_1^{\text{macro}}\left(Y_{\text{test}}, g_L^*(Z_{\text{test}})\right).$$

The star denotes validation selection within the pre-specified linear-probe family and hyperparameter grid.

A.5 RELATIONSHIP AMONG THE THREE READOUTS

The three readouts form two controlled comparisons.

First, R_P and A_P retain the same one-prototype-per-class Euclidean decision form but differ in whether TRAIN labels determine prototype locations. Their difference,

$$\Delta_{\text{disc}} = A_P - R_P,$$

therefore isolates the accessibility gained from supervised prototype placement.

Second, moving from A_P to $D_L^\star$ relaxes the one-centroid constraint. Nearest-centroid prediction can be written as

$$\begin{aligned}\arg\min_c \|z - \mu_c\|_2^2 &= \arg\min_c \left(\|z\|_2^2 - 2\mu_c^\top z + \|\mu_c\|_2^2\right) \\ &= \arg\max_c \left(2\mu_c^\top z - \|\mu_c\|_2^2\right).\end{aligned}$$

Hence nearest-centroid classification is a restricted affine multiclass rule, whereas $D_L^\star$ is selected from a broader controlled affine family.

This hypothesis-space relationship is used to interpret readout constraints, not to impose a strict empirical ordering among R_P, A_P, and $D_L^\star$. R_P depends on a finite K-Means solution and $D_L^\star$ on a finite validation-selected probe family, so small empirical departures from a capacity-based ordering are diagnostic rather than logical contradictions.

B EVALUATION AND STATISTICAL PROTOCOL

B.1 REPRESENTATION RUNS AND DATA USAGE

Unless otherwise stated, experiments use five data splits and three model seeds per split, yielding

$$5 \times 3 = 15$$

trained representation runs for each dataset–architecture setting. Models compared within a dataset use the same split definitions.

TRAIN, VAL, and TEST roles are fixed throughout. TRAIN nodes determine standardization statistics, K-Means prototypes, cluster–class mappings, supervised class centroids, and fitted probe parameters. VAL is used only when a procedure requires model or hyperparameter selection, including selection within the linear-probe family. TEST labels are reserved for final evaluation.

No TEST information is used to refit standardization, relocate prototypes, remap clusters, construct supervised centroids, or select linear probes.

B.2 K-MEANS REPETITIONS

Metrics involving K-Means use fixed external repetitions to reduce dependence on clustering initialization. All repetitions belonging to a trained representation are aggregated before statistical inference.

K-Means repetitions are therefore repeated measurements of the same frozen representation rather than independent representation samples. Statistical uncertainty is estimated across data splits and independently trained model seeds, not by treating clustering initializations as additional model runs.

The same principle applies to prototype-budget and tessellation analyses: external clustering repetitions characterize within-run clustering variability and are collapsed before cross-run inference.

B.3 HIERARCHICAL UNCERTAINTY ESTIMATION

Uncertainty is estimated using a two-stage split–model-seed hierarchical bootstrap. The data split forms the outer sampling level. Within each sampled split, model seeds are resampled at the second level.

For a scalar quantity M, a bootstrap replicate therefore has the schematic form

$$\widehat{M}^{(b)} = \mathrm{Agg}\left\{M_{s^*,m^*}\right\},$$

where s^* denotes a resampled data split and m^* a model seed resampled within that split.

Architecture comparisons are paired at the data-split level because the architectures use identical split definitions. Model seeds from different architectures are not treated as one-to-one paired random replicates; they are resampled independently within each paired split.

Unless otherwise stated, reported confidence intervals are 95% hierarchical bootstrap intervals after all within-run K-Means repetitions have been aggregated. Exact directional split-level tests are used only as supporting evidence and do not replace the hierarchical uncertainty estimates.

B.4 INFERENTIAL SCOPE

Measurements on already-trained frozen representations are observational. They can establish accessibility profiles, localize a geometric failure, and identify recurring structural associations, but they do not by themselves identify causal mediation.

For intervention experiments, target-degree message scaling is modified during training and the representation is retrained under each treatment. Causal interpretation is restricted to the effect of this operator-level training manipulation on the representation learned under that treatment.

Representation-level quantities measured after treatment, including G_{degree} and other geometric diagnostics, are post-treatment measurements. Their co-variation with accessibility is not interpreted as identified causal mediation unless a separate mediation design establishes that relationship.

C LOCALIZATION AND PROTOTYPE-BUDGET DIAGNOSTICS

All localization measurements operate on frozen representations. Projection bases, prototypes, and cluster–class mappings are constructed using TRAIN data only; TEST labels are used only for final evaluation.

C.1 CLASS-CENTROID SUBSPACE

For class c, define the TRAIN centroid

$$\mu_c = \frac{1}{|V_c^{\mathrm{train}}|} \sum_{i \in V_c^{\mathrm{train}}} z_i, \qquad \bar{\mu} = \frac{1}{C} \sum_{c=1}^{C} \mu_c.$$

The class-centroid subspace is

$$U_C = \mathrm{span}\left\{\mu_1 - \bar{\mu}, \ldots, \mu_C - \bar{\mu}\right\},$$

whose dimension is at most $C - 1$. Any representation decomposes as

$$z = P_{U_C} z + P_{U_C^\perp} z.$$

$U_C^\perp$ is not interpreted as a non-semantic subspace. The decomposition tests only whether centroid-orthogonal directions are necessary for the prototype-recovery deficit.

For the original space, U_C, and $U_C^\perp$, the complete recovery procedure is refitted independently for each architecture. K-Means is fitted on projected TRAIN embeddings, the cluster–class mapping is learned from TRAIN labels, and both prototypes and mappings are then frozen for TEST prediction. No prototype or mapping fitted in one space is inherited by another.

For GraphSAGE on Cora, projection onto U_C leaves all three accessibility readouts virtually unchanged:

	Full space	U_C	U_C-Full
R_P	0.8431	0.8431	0.0000
A_P	0.8429	0.8429	0.0000
$D_L^\star$	0.8454	0.8465	+0.0010

R_P and A_P are identical across all 15 runs; the $D_L^\star$ change is within run-level noise (mean absolute change 0.0051, maximum 0.0183). This confirms that the projection procedure itself does not induce a discovery deficit, and that the GIN-specific failure is not an artifact of projecting onto U_C.

C.2 Prototype-budget intervention

Prototype capacity is varied according to

$$K = qC, \qquad q \in \{1, 2, 3, 4\}.$$

When $q > 1$, multiple discovered clusters may correspond to the same class, so each cluster is assigned the majority class among its TRAIN members. The same TRAIN-only many-to-one mapping rule is applied at $q = 1$. Hence the contrast between $q = 1$ and $q > 1$ changes prototype capacity while holding the mapping criterion fixed.

For each q, K-Means is refitted on TRAIN embeddings and the resulting prototypes and mapping are frozen before TEST prediction. External K-Means repetitions are aggregated within each representation run before cross-run inference.

Within U_C, GIN TEST Macro-F1 increases from approximately 0.375 at $q = 1$ to 0.776 at $q = 4$, whereas GraphSAGE changes from approximately 0.843 to 0.850. The response is interpreted as evidence that one discovered Euclidean prototype per class is an insufficient readout model for the frozen GIN representation. It does not identify the number or semantic meaning of latent within-class modes.

C.3 Supervised–unsupervised tessellation mismatch

Inside U_C, let the supervised nearest-centroid assignment of TEST node z_i be

$$\hat{v}_i = \arg\min_c \left\| P_{U_C} z_i - \mu_c \right\|_2 ,$$

and let its assignment under standard $K = C$ K-Means be

$$\hat{k}_i = \arg\min_k \left\| P_{U_C} z_i - \hat{\mu}_k \right\|_2 .$$

For supervised centroid region c, define

$$p_{ck} = P(K = k \mid V = c).$$

The entropy-effective number of contributing discovered regions is

$$N_{\text{eff}}(c) = \exp\left[- \sum_{k : p_{ck} > 0} p_{ck} \log p_{ck} \right],$$

and the dominant-region mass is

$$M_{\mathrm{dom}}(c) = \max_k p_{ck}.$$

$N_{\mathrm{eff}} = 1$ indicates that a supervised centroid region is concentrated within one discovered K-Means region; larger values indicate that its mass is distributed across multiple discovered regions. M_{dom} measures the fraction captured by the dominant discovered region.

The GIN–GraphSAGE architecture contrasts are

$$\Delta N_{\mathrm{eff}} = 0.4990, \qquad 95\%\ \mathrm{CI} = [0.4492, 0.5482],$$

and

$$\Delta M_{\mathrm{dom}} = -0.1594, \qquad 95\%\ \mathrm{CI} = [-0.1765, -0.1420].$$

These quantities measure partition mismatch under the standard $q = 1$ setting. They do not estimate a number of stable semantic subclusters.

C.4 Inferential status

Projection, prototype-budget, and tessellation measurements are performed on already-trained frozen representations and therefore support localization and geometric characterization rather than causal claims.

These analyses characterize mechanism rather than serving as preregistered confirmatory gates. Only the intervention endpoints in Section 5 were preregistered.

D Degree-Conditioned Diagnostics and Cross-Dataset Recurrence

This appendix gives the full definitions and controls underlying Section 4. All measurements operate on frozen deterministic representations; no intervention is performed in this appendix.

D.1 Class-conditional degree strata

Node degree is defined on the canonical simple-undirected graph. Because raw degree distributions can differ across classes, degree strata are constructed within each true class using TRAIN-node quantiles and then frozen on TEST. Thus, a stratum index denotes relatively low, medium, or high degree with respect to that class's own TRAIN distribution rather than an absolute degree range shared across classes.

D.2 Partition-level diagnostic

For a standard $q = 1$, $K = C$ K-Means fitted on TRAIN, let K denote TEST-node prototype assignments and let D_{bin} denote the frozen class-conditional degree stratum. We measure

$$I_{\mathrm{adj}}(K; D_{\mathrm{bin}} \mid Y),$$

where the observed conditional mutual information is centered by a within-class permutation null that preserves the class-specific marginal counts. External K-Means repetitions are first averaged within each representation run and are not treated as independent statistical samples. This statistic asks whether degree stratum provides information about which prototype a node occupies beyond what is already explained by the true class.

D.3 Geometry-level diagnostic

To avoid relying on the K-Means partition, we also measure degree-conditioned geometry directly inside the TRAIN-centroid subspace U_C. Let n_{cb} and $\bar{z}^C_{cb}$ denote the number and centroid of TEST nodes in true class c and degree stratum b, and let $\bar{z}^C_c$ denote the overall TEST centroid of class c. We define

$$G_{\mathrm{degree}} = \frac{\sum_{c \in \mathcal{C}_s} \sum_{b: n_{cb} > 0} n_{cb} \left\| \bar{z}^C_{cb} - \bar{z}^C_c \right\|_2^2}{\sum_{c \in \mathcal{C}_s} \sum_{i \in T_c} \left\| z^C_i - \bar{z}^C_c \right\|_2^2}.$$

This ratio measures the fraction of pooled within-class squared scatter attributable to mean shifts between class-conditional degree strata. It is independent of K-Means assignments, cluster mappings, and clustering seeds. On Cora, the GIN–GraphSAGE contrasts are $\Delta I_{\mathrm{adj}} = 0.10705$ (95% CI $[0.07843, 0.13711]$) and $\Delta G_{\mathrm{degree}} = 0.24493$ (95% CI $[0.21217, 0.28015]$).

D.4 Supporting four-architecture Cora characterization

As a supporting breadth analysis, we applied the same fixed-dose $q = 4$ prototype-capacity and degree-conditioned diagnostics to GIN, GCN, GATv2 (Brody et al., 2022), and GraphSAGE on Cora. This panel is descriptive: it does not rank architectures by intrinsic "degree exposure," and a confidence interval containing zero is not interpreted as evidence of absence or equivalence.

Encoder	$\Delta^{U_C}_{q=4}$	I_{adj}	G_{degree}
GIN	$\mathbf{0.4016}$ $[\mathbf{0.3623}, \mathbf{0.4405}]$	$\mathbf{0.1084}$ $[\mathbf{0.0807}, \mathbf{0.1359}]$	$\mathbf{0.2662}$ $[\mathbf{0.2351}, \mathbf{0.2989}]$
GCN	0.0494 $[0.0248, 0.0834]$	0.0074 $[0.0012, 0.0138]$	0.0746 $[0.0664, 0.0826]$
GATv2	0.0048 $[-0.0015, 0.0112]$	0.0022 $[-0.0026, 0.0079]$	0.0222 $[0.0154, 0.0293]$
GraphSAGE	0.0072 $[0.0030, 0.0112]$	0.0013 $[-0.0040, 0.0083]$	0.0212 $[0.0169, 0.0255]$

All three metrics place GIN highest, followed by GCN, with GATv2 and GraphSAGE substantially smaller. The corresponding GIN-minus-comparator contrasts have positive 95% hierarchical-bootstrap confidence intervals for all three metrics.

Two points about interpretation should be kept explicit. First, the $\Delta^{U_C}_{q=4}$ column reports the fixed-dose characterization described in Appendix C, not the validation-selected Δ_{over} used in Table 3. Second, GATv2 introduces degree dependence through attention normalization rather than through an explicit degree scaler, so its G_{degree} value measures degree-conditioned organization of the learned representation rather than the explicit degree dependence of its operator.

D.5 Amazon-Computers recurrence protocol

Amazon-Computers was selected using only its previously observed accessibility phenotype. The prototype-budget and degree-conditioned outcomes below were not used for dataset selection, so we describe this design as phenotype-conditioned recurrence rather than blind replication.

For the prototype-capacity analysis, we fit $K = qC$ prototypes for $q \in \{1, 2, 3, 4\}$. For $q > 1$, multiple clusters must be allowed to map to the same class. To separate the effect of this mapping relaxation from the effect of adding prototypes, we define

$$\Delta_{\mathrm{map}} = R_{q=1,\mathrm{m2o}} - R_P, \qquad \Delta_{\mathrm{over}} = R_{q^*,\mathrm{m2o}} - R_{q=1,\mathrm{m2o}},$$

where $q^* \in \{2, 3, 4\}$ is selected using validation Macro-F1 only; TEST is never used for the selection. Section 4 reports Δ_{over} because it isolates the gain attributable to additional prototype capacity rather than to changing the cluster–class mapping rule.

On Amazon-Computers, the additional-prototype gain is 0.3535 for GIN and 0.2071 for GraphSAGE. The corresponding degree-conditioned contrasts are $\Delta I_{\mathrm{adj}} = 0.21662$ (95% CI $[0.18445, 0.24678]$) and $\Delta G_{\mathrm{degree}} = 0.20282$ (95% CI $[0.15154, 0.25645]$). These recurrence measurements are observational and were defined independently of the subsequent intervention analysis; they support recurrence of the antecedent pattern rather than causal replication. Table 3 summarizes the deterministic accessibility profiles and degree-conditioned diagnostics for both datasets.

Here, Δ_{over} denotes the additional-prototype gain relative to $q = 1$ under the unified many-to-one mapping, with $q^\star \in \{2, 3, 4\}$ selected using validation Macro-F1 within each representation run. Both Cora and Amazon-Computers use this validation-selected estimand. The fixed-$q = 4$ curves in Section 3.2 are reported separately as a controlled prototype-dose characterization.

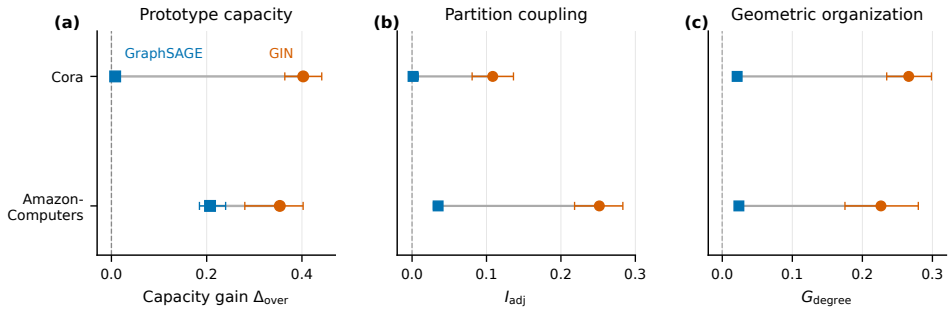


Figure 4: **Prototype-capacity and degree-conditioned recurrence.** Rows show Cora and Amazon-Computers; columns show **(a)** the additional-prototype recovery gain Δ_{over}, **(b)** degree-conditioned partition coupling I_{adj}, and **(c)** degree-conditioned geometric organization G_{degree}. Orange circles denote GIN and blue squares denote GraphSAGE. Horizontal bars show 95% confidence intervals for the architecture endpoints; intervals in (b,c) are descriptive. Gray lines connect architecture means within each dataset and do not indicate paired model seeds. Each column has its own horizontal scale. Amazon-Computers was selected using a previously frozen discovery-dominant phenotype.

Table 3: Accessibility profiles and degree-conditioned organization across datasets and encoders. K-Means repetitions are aggregated within each representation run before summarizing over five data splits and three model seeds per split.

Dataset	Encoder	R_P	A_P	$D_L^\star$	Δ_{over}	I_{adj}	G_{degree}
Cora	GIN	**0.3863**	0.7592	**0.8511**	0.4027	0.1084	0.2662
Cora	GraphSAGE	**0.8431**	**0.8429**	**0.8454**	0.0079	0.0013	0.0212
Amazon-Computers	GIN	0.3299	0.6908	0.8575	0.3535	0.2516	0.2265
Amazon-Computers	GraphSAGE	0.6586	0.8386	0.8758	0.2071	0.0350	0.0237

E Target-Degree Intervention Details

E.1 GIN target-degree attenuation

Let d_v denote the degree of target node v in the canonical simple-undirected graph and define

$$\widetilde{d}_v = \max(d_v, 1).$$

Native GIN updates node v as

$$h_v^{(\ell+1)} = \text{MLP}_\ell \left[(1+\epsilon_\ell) h_v^{(\ell)} + \sum_{u \in \mathcal{N}(v)} h_u^{(\ell)} \right].$$

The attenuation intervention modifies only the neighborhood-sum branch:

$$h_v^{(\ell+1)} = \text{MLP}_\ell \left[(1+\epsilon_\ell) h_v^{(\ell)} + \widetilde{d}_v^{-\alpha} \sum_{u \in \mathcal{N}(v)} h_u^{(\ell)} \right].$$

Thus, the multiplier is applied after neighborhood summation but before the neighborhood term is combined with the self term and passed through the native GIN MLP. The self contribution

$$(1+\epsilon_\ell) h_v^{(\ell)}$$

is not scaled, and the MLP placement is unchanged.

For non-isolated nodes,

$$\sum_{u \in \mathcal{N}(v)} h_u^{(\ell)} = d_v \, \bar{h}_{\mathcal{N}(v)}^{(\ell)},$$

so the intervened neighborhood term can be written as

$$d_v^{1-\alpha}\bar{h}_{\mathcal{N}(v)}^{(\ell)}$$

up to the use of $\widetilde{d}_v$ for isolated nodes. The explicit target-degree exponent therefore changes from 1 at $\alpha = 0$ toward 0 at $\alpha = 1$.

The Cora intervention grid is

$$\alpha \in \{0, 0.25, 0.5, 0.75, 1\},$$

whereas Amazon-Computers uses

$$\alpha \in \{0, 0.5, 1\}.$$

At $\alpha = 0$, the intervention path reduces exactly to native GIN. At $\alpha = 1$, the neighborhood summary becomes mean-like, but the resulting operator is not GraphSAGE because the GIN self term, MLP structure, and remaining architecture-specific computation are unchanged.

A simple scaling calculation provides intuition for the intervention without constituting a model of the learned embedding. Ignoring the self term, nonlinearity, and training adaptation, suppose neighbor messages have conditional mean μ_y and covariance Σ_y given class y and target degree d. Then

$$\mathbb{E}[a_v^{(\alpha)} \mid y, d] = d^{\,1-\alpha}\mu_y,$$

and

$$\mathrm{Cov}\left[a_v^{(\alpha)} \mid y, d\right] = d^{\,1-2\alpha}\Sigma_y.$$

This calculation motivates α as a direct manipulation of explicit target-degree scaling in the neighborhood contribution; it does not imply a monotonic response of the learned representation or its accessibility measures.

E.2 GRAPHSAGE TARGET-DEGREE AMPLIFICATION

For mean-aggregating GraphSAGE, the native neighborhood representation is

$$m_v^{(\ell)} = \mathrm{Mean}_{u\in\mathcal{N}(v)}\, h_u^{(\ell)} = \bar{h}_{\mathcal{N}(v)}^{(\ell)}.$$

The amplification intervention scales only this neighborhood branch:

$$\widetilde{m}_v^{(\ell)}(\beta) = \widetilde{d}_v^{\beta} m_v^{(\ell)}.$$

The resulting layer update is

$$h_v^{(\ell+1)} = \sigma\Big(W_{\mathrm{self}}^{(\ell)} h_v^{(\ell)} + W_{\mathrm{neigh}}^{(\ell)} \widetilde{m}_v^{(\ell)}(\beta) + b^{(\ell)}\Big).$$

The degree multiplier is therefore applied after mean aggregation and before the learned neighborhood transform $W_{\mathrm{neigh}}^{(\ell)}$. The root branch

$$W_{\mathrm{self}}^{(\ell)} h_v^{(\ell)}$$

is left unchanged. Consequently, increasing β changes the degree-dependent scale of the neighborhood contribution relative to the fixed root contribution; it does not multiply the complete GraphSAGE layer output.

The frozen intervention grid is

$$\beta \in \{0, 0.5, 1\}.$$

At $\beta = 0$, the intervention path reduces exactly to native GraphSAGE. For non-isolated nodes, $\beta = 1$ gives

$$\widetilde{m}_v^{(\ell)}(1) = d_v\, \bar{h}_{\mathcal{N}(v)}^{(\ell)},$$

so the pre-transform neighborhood summary has the same explicit target-degree prefactor d_v^1 as a neighborhood sum. This does not make the resulting GraphSAGE layer equivalent to GIN because the separate root and neighborhood transforms, activation, and other architecture-specific components remain those of GraphSAGE.

E.3 Relation between the two intervention families

At the level of the pre-transform neighborhood summary, the two interventions move explicit target-degree dependence in opposite directions:

$$\text{GIN:} \qquad d_v^1 \xrightarrow{\alpha\uparrow} d_v^0,$$

whereas

$$\text{GraphSAGE:} \qquad d_v^0 \xrightarrow{\beta\uparrow} d_v^1.$$

This correspondence concerns only the explicit target-degree prefactor of the neighborhood summary. It does not make the intervention endpoints architecture-equivalent:

$$\alpha = 1 \not\equiv \text{GraphSAGE}, \qquad \beta = 1 \not\equiv \text{GIN}.$$

Likewise, α and β are not quantitatively matched treatment doses. They are opposite-direction manipulations of a common operator-level scaling dimension within two different message-passing architectures.

E.4 Paired treatment design and causal estimand

Each intervention is evaluated relative to the native baseline of the same architecture. Experiments use five data splits and three model seeds per split, yielding

$$5 \times 3 = 15$$

paired treatment cells for each dataset–architecture setting.

Within a split–seed cell, treatment values share the same data split, initialization, training policy, and deterministic execution settings. Treatment contrasts are therefore formed within cell before aggregation across splits and model seeds. Statistical uncertainty follows the split–model-seed hierarchical procedure in Appendix B.

The native endpoints are implementation controls rather than separate architectures: $\alpha = 0$ is required to reproduce native GIN and $\beta = 0$ native GraphSAGE under their respective intervention paths. Operator-identity and intervention-integrity checks verify that the intended target-degree multiplier is the treatment change.

Because α and β modify the forward operator during training, each non-native treatment learns a new representation. The intervention effect therefore includes both the direct change in message scaling and the trained model's adaptation to that operator. The causal estimand is the effect of the training-time operator intervention on the representation learned under that treatment and its subsequent accessibility profile; it is not the effect of a post-hoc transformation applied to a fixed embedding.

E.5 GIN clean-repair test

The intervention grid $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and the clean-repair endpoints were frozen before inspecting $\alpha > 0$ outcomes. Besides G_{degree} and $R_P^{U_C}$, the partition endpoint is

$$H_{\text{norm}}(K \mid V) = \frac{H(K \mid V)}{\log K},$$

where K is the standard $q = 1$ K-Means assignment and V is the supervised centroid Voronoi cell. The preregistered prediction required G_{degree} and $H_{\text{norm}}(K \mid V)$ to decrease, $R_P^{U_C}$ to increase, and both A_P and $D_L^\star$ to satisfy their non-inferiority margins. The conjunction fails if any endpoint fails.

E.6 GIN linear-probe selection sensitivity

For each treatment representation, $D_L^\star$ reselects the best member of the same pre-frozen probe family and hyperparameter grid using that treatment's VAL set, then evaluates TEST once. To test whether the endpoint loss is caused by discrete selection switching, we also freeze the family and hyperparameter chosen at $\alpha = 0$ and refit that same configuration on every treatment representation. This $D_L^{\text{fixed-config0}}$ endpoint changes by -0.1269 (95% CI $[-0.1825, -0.0716]$), compared with -0.1066 (95% CI $[-0.1548, -0.0573]$) for $D_L^\star$. The common direction shows that the linear-accessibility loss is not explained by probe-selection switching alone.

F POST-INTERVENTION GEOMETRIC AUDIT

This appendix provides post-treatment structural diagnostics. The first subsection compares accessibility and G_{degree} trajectories across intervention settings; subsequent subsections audit the Cora–GIN attenuation geometry in detail. These analyses are explanatory rather than mediational: the quantities below are not interpreted as identified causal mediators of prototype, centroid, or linear accessibility.

F.1 ACCESSIBILITY AND DEGREE-ORGANIZATION TRAJECTORIES

The intervention trajectories provide an additional check on whether G_{degree} can be interpreted as a scalar mediator of accessibility. Because G_{degree} is measured after retraining under each treatment, these comparisons are descriptive post-treatment diagnostics rather than a formal mediation analysis.

For Cora–GraphSAGE, the midpoint treatment already produces a substantial change in degree-conditioned organization:

$$G_{\mathrm{degree}}: \qquad 0.021 \to 0.146 \to 0.233$$

for

$$\beta = 0,\ 0.5,\ 1.$$

Relative to the total endpoint change, the midpoint accounts for approximately

$$\frac{0.146 - 0.021}{0.233 - 0.021} \approx 0.59$$

of the observed increase in G_{degree}.

Prototype accessibility does not follow the same trajectory:

$$R_P^{U_C}: \qquad 0.843 \to 0.830 \to 0.395.$$

The midpoint therefore accounts for only approximately

$$\frac{0.843 - 0.830}{0.843 - 0.395} \approx 0.03$$

of the eventual endpoint decrease in prototype recovery. Thus, a large fraction of the observed change in degree-conditioned organization appears before the large discovery-side accessibility shift.

Amazon-Computers provides a complementary mismatch. Under GIN attenuation, the midpoint treatment changes

$$R_P^{U_C}: \qquad 0.336 \to 0.532,$$

and

$$A_P: \qquad 0.691 \to 0.827,$$

even though

$$G_{\mathrm{degree}}: \qquad 0.227 \to 0.247$$

does not decrease at $\alpha = 0.5$. At the endpoint, G_{degree} subsequently decreases to approximately 0.022.

The two trajectories therefore show that G_{degree} and accessibility need not change proportionally, or even in the same temporal order along the treatment grid. These observations are inconsistent with a simple scalar-mediation account in which accessibility is determined monotonically by the measured value of G_{degree}.

They do not exclude more complex causal relations involving degree-conditioned organization. In particular, the treatment changes the message-passing operator and allows the learned representation to adapt jointly, while G_{degree} is measured only after treatment. We therefore use G_{degree} as a post-treatment structural diagnostic and do not interpret it as an identified causal mediator.

F.2 Treatment-Specific Centroid Geometry

For each normalization strength α, we construct the class-centroid subspace from the corresponding retrained representation. Let $\mu_c^{(\alpha),\text{train}}$ denote the TRAIN centroid of class c under treatment α, and define

$$\bar{\mu}^{(\alpha)} = \frac{1}{C}\sum_{c=1}^{C} \mu_c^{(\alpha),\text{train}}.$$

The treatment-specific class-centroid subspace is

$$U_C^{\alpha} = \text{span}\left\{\mu_1^{(\alpha),\text{train}} - \bar{\mu}^{(\alpha)}, \ldots, \mu_C^{(\alpha),\text{train}} - \bar{\mu}^{(\alpha)}\right\}.$$

Each treatment is therefore analyzed in its own centroid coordinate system, rather than by projecting all treatment representations onto the centroid subspace of the untreated model. This distinction matters because the intervention changes both the learned representation and its class-centroid geometry.

For the class-conditional radial audit, each TEST node is centered at its own TRAIN class centroid before projection. For a TEST node i with true class y_i, we define the centroid-orthogonal radius as

$$r_{\perp,i}^{(\alpha)} = \left\| P_{(U_C^{\alpha})^{\perp}} \left(z_i^{(\alpha)} - \mu_{y_i}^{(\alpha),\text{train}} \right) \right\|_2.$$

Thus, $r_{\perp,i}^{(\alpha)}$ is the Euclidean norm of the class-centered orthogonal residual; it is neither a coordinate-wise magnitude nor a radius around the global training mean.

The reported upper-quantile statistic is computed within each true class and then macro-averaged:

$$Q_{0.90}(r_{\perp}) = \frac{1}{C}\sum_{c=1}^{C} Q_{0.90}\left(r_{\perp,i}^{(\alpha)} \mid i \in \text{TEST},\ y_i = c \right).$$

For the full-grid summary curve, these run-level class-macro values are then averaged over the 5 data splits and 3 model seeds per split. Statistical uncertainty follows the split–model-seed hierarchical procedure described in Appendix B.

F.3 Shared Contraction of the Centroid-Orthogonal Component

The most stable treatment response occurs in the component orthogonal to U_C^{α}. Across $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$, the macro-averaged upper quantile follows

$$Q_{0.90}(r_{\perp}): \quad 1.6075 \to 1.3949 \to 0.8318 \to 0.4535 \to 0.1578.$$

The endpoint change has the same direction across all seven classes.

This result establishes that the operator-level intervention substantially contracts a broadly shared centroid-orthogonal component. Importantly, Section 3 showed that this component is not required for the single-prototype discovery failure: projecting the original GIN representation into the supervised centroid subspace preserves most of the failure and of the multi-prototype rescue. The present result therefore does not identify centroid-orthogonal spread as the cause of poor prototype recovery. Instead, it shows that changing target-degree dependence during training has a strong and systematic effect on this component of the learned geometry.

F.4 Degree–Radius Organization and Relative Radial Shape

We separately measure whether node degree orders the full class-centered radial geometry. For each true class, we compute the Spearman correlation between log-degree and the Euclidean distance to the corresponding TRAIN class centroid, then macro-average across classes:

$$\rho_{\text{degree,radius}} = \frac{1}{C}\sum_{c=1}^{C} \rho_S\left(\log(1+d_i), \left\| z_i^{(\alpha)} - \mu_c^{(\alpha),\text{train}} \right\|_2 \right), \qquad i \in \text{TEST},\ y_i = c.$$

This statistic uses the full class-centered radius, not the centroid-orthogonal radius $r_{\perp}$.

Table 4: Per-class endpoint changes under degree normalization ($\Delta = \alpha{=}1 - \alpha{=}0$). $r_\perp$ denotes the class-centered centroid-orthogonal radius, r the full class-centered radius, and separation the distance to the nearest competing TRAIN class centroid. Values are mean paired endpoint changes with 95% split–model-seed hierarchical-bootstrap confidence intervals.

Class	$\Delta Q_{0.90}(r_\perp)$	$\Delta[Q_{0.90}(r)/Q_{0.50}(r)]$	Δ nearest-centroid separation	$\Delta[Q_{0.90}(r)/\text{separation}]$	Δ centroid violation
0	$-1.4689[-2.0120, -0.9659]$	$+0.0722[-0.4728, +0.5903]$	$-0.7909[-2.3256, +0.9515]$	$+0.0049[-0.3835, +0.3791]$	$+0.0423[-0.0526, +0.1333]$
1	$-1.6536[-2.0069, -1.2803]$	$+0.9119[+0.4129, +1.3780]$	$-3.5775[-4.9332, -2.3720]$	$+1.5287[+0.0512, +3.3149]$	$+0.0551[-0.0818, +0.2081]$
2	$-1.2544[-1.5478, -0.9582]$	$+0.5100[-0.0465, +1.0864]$	$-2.6739[-4.0828, -1.1265]$	$+0.1236[-0.0548, +0.3247]$	$+0.0169[-0.0313, +0.0594]$
3	$-1.2277[-1.5489, -0.9202]$	$+0.6223[+0.3070, +0.9153]$	$+6.1588[+5.0901, +7.2694]$	$-0.0570[-0.1651, +0.0428]$	$-0.0163[-0.0675, +0.0245]$
4	$-1.4866[-2.0042, -1.0760]$	$+1.0163[+0.8089, +1.2329]$	$+1.6684[-0.0595, +2.8845]$	$-0.3004[-0.4314, -0.1777]$	$-0.1574[-0.2062, -0.1109]$
5	$-1.4546[-1.6334, -1.2547]$	$+0.7962[+0.5054, +1.1290]$	$-2.6524[-3.9363, -1.3419]$	$+2.4965[+0.6199, +5.6444]$	$+0.1820[+0.0804, +0.2844]$
6	$-1.6016[-1.9936, -1.2098]$	$+0.6779[+0.2324, +1.1427]$	$-2.6625[-3.9315, -1.2829]$	$+1.5274[+0.2757, +3.5672]$	$+0.2241[+0.0537, +0.3852]$

Its macro-averaged trajectory is

$$\rho_{\text{degree,radius}}: \quad 0.1341 \to 0.1377 \to 0.1356 \to 0.0418 \to -0.1261.$$

The association is therefore essentially unchanged through moderate normalization ($\alpha \leq 0.5$) and changes strongly only under larger α. This distinction is important: by $\alpha = 0.5$, $Q_{0.90}(r_\perp)$ has already fallen from 1.6075 to 0.8318, whereas $\rho_{\text{degree,radius}}$ remains nearly unchanged ($0.1341 \to 0.1356$). Hence, centroid-orthogonal amplitude contraction is not equivalent to the disappearance of degree–radius ordering.

At the same time, normalization does not simply make every class distribution more spherical or homogeneous. The relative radial span

$$\frac{Q_{0.90}}{Q_{0.50}}$$

increases from approximately 2.079 at $\alpha = 0$ to 2.737 at $\alpha = 1$. This increase should not be interpreted as heavier absolute tails. Standardized skewness and excess kurtosis decrease over the same intervention. A more precise interpretation is that the central mass contracts faster than the upper radial extent, increasing relative radial heterogeneity even while the overall representation contracts.

Taken together, these measurements separate three aspects of the treatment response: centroid-orthogonal amplitude contracts broadly across classes, full-radius degree ordering changes mainly under stronger normalization, and relative radial shape follows yet another dose response.

F.5 Class-Selective Centroid Reconfiguration

The shared contraction does not imply a uniform change in supervised centroid geometry. To characterize class-specific effects, we jointly inspect within-class radial extent, nearest competing-centroid separation, and the fraction of nodes violating the nearest-centroid decision geometry. A useful summary quantity is the relative geometric occupancy

$$\frac{\text{within-class extent}}{\text{available centroid separation}}.$$

A class may therefore contract in absolute radius while becoming less compatible with a nearest-centroid readout if its available centroid separation contracts even faster.

Class 4: favorable centroid reconfiguration. Class 4 becomes more compatible with a single-centroid readout as normalization increases. Its normalized radial-to-separation measure decreases from approximately 1.242 to 0.941, while its centroid-violation rate decreases from 0.351 to 0.193. The fraction of nodes that are incorrect under the centroid readout but recovered by the linear readout also decreases from approximately 0.226 to 0.033. This pattern indicates an improvement of centroid geometry itself rather than merely a disappearance of linearly accessible information.

Table 4 reports the corresponding endpoint changes for all seven classes, showing that the shared orthogonal contraction is universal whereas the centroid-level response is heterogeneous.

Class 5: contraction with relative centroid compression. Class 5 exhibits the opposite behavior. Its within-class radial scale contracts: the median radius decreases from approximately 2.99 to 1.77,

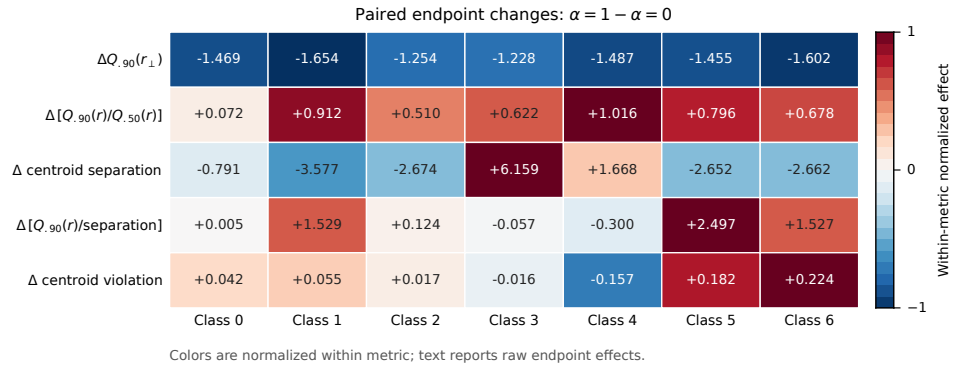


Figure 5: **Degree normalization combines shared centroid-orthogonal contraction with class-selective centroid reconfiguration.** Cells report the endpoint effect $\alpha = 1 - \alpha = 0$ for each Cora class. All seven classes show contraction in the upper quantile of the centroid-orthogonal radius $Q_{0.90}(r_\perp)$, whereas relative radial geometry, nearest-centroid separation, and centroid-violation rates respond heterogeneously across classes. In particular, Class 4 becomes more compatible with a nearest-centroid readout, while Classes 5–6 lose relative centroid separation despite contracting in absolute radius. Colors are normalized within each metric for visualization; cell annotations report the raw endpoint effects. These quantities are post-intervention explanatory correlates rather than identified causal mediators.

and its upper radial quantile decreases from approximately 6.40 to 5.00. However, nearest-centroid separation decreases more strongly, from approximately 5.62 to 2.97. Consequently, its relative radial-to-separation geometry deteriorates, and the centroid-violation rate increases from 0.164 to 0.346. Thus, absolute within-class contraction is not sufficient for improved centroid accessibility.

Additional class-level responses. Class 6 exhibits the same qualitative failure mode as Class 5: its radial spread contracts, but centroid separation contracts even more strongly, and its centroid-violation rate increases from approximately 0.163 to 0.387. The response is not universally degradative, however. Class 3 expands in absolute radial extent, but its centroid separation increases more strongly, yielding a slight improvement in relative centroid geometry and a decrease in centroid violation from approximately 0.226 to 0.210. Classes 0 and 2 are comparatively stable in centroid geometry apart from the shared centroid-orthogonal contraction.

Figure 5 summarizes the endpoint pattern: the centroid-orthogonal contraction is shared across classes, whereas changes in centroid geometry are strongly class-selective.

$$\text{shared centroid-orthogonal contraction} \quad + \quad \text{class-selective centroid reconfiguration.}$$

The first effect is directionally consistent across classes; the second may improve, degrade, or leave approximately unchanged the geometry available to a supervised centroid readout.

F.6 Dose Alignment with the Accessibility Response

The geometric audit also helps characterize the non-monotonic accessibility response reported in the main text.

Over the moderate-normalization interval, from $\alpha = 0$ to $\alpha = 0.5$, the centroid-orthogonal upper radius already decreases substantially:

$$Q_{0.90}(r_\perp): \quad 1.6075 \to 0.8318.$$

Over the same interval, prototype recovery increases from approximately 0.390 to 0.549, while supervised centroid accessibility remains approximately preserved ($A_P : 0.759 \to 0.779$).

This dose alignment is consistent with an early regime in which a substantial centroid-orthogonal contraction co-occurs with improved prototype recovery before major centroid degradation appears. Notably, the full-radius degree–radius rank association remains almost unchanged over the same

range, from 0.1341 to 0.1356. The early contraction therefore should not be described as the disappearance of degree–radius organization.

Beyond $\alpha = 0.5$, the centroid-orthogonal component continues to contract,

$$0.8318 \to 0.4535 \to 0.1578,$$

while the degree–radius association changes much more strongly,

$$0.1356 \to 0.0418 \to -0.1261.$$

Prototype recovery no longer improves,

$$0.549 \to 0.537 \to 0.509.$$

At the same time, several classes—most clearly Classes 5 and 6—develop strong relative centroid compression, while supervised centroid and linear accessibility decline.

A useful descriptive summary is therefore

$$\underbrace{\text{shared centroid-orthogonal contraction}}_{\text{co-occurring with early recovery gain}} \quad + \quad \underbrace{\text{class-selective centroid reconfiguration}}_{\text{co-occurring with later accessibility loss}}.$$

The full-radius degree ordering changes mainly in the stronger-normalization regime and should be treated as a distinct measured response rather than as the mechanism underlying the earlier orthogonal contraction.

This summary describes the observed treatment trajectories; it is not a quantitative mediation decomposition.

F.7 Interpretation and Claim Boundary

The geometric audit is inconsistent with a simple interpretation in which stronger degree normalization merely removes a uniformly harmful nuisance component. Prototype, centroid, and linear accessibility exhibit distinct dose responses, while class-level centroid geometry reorganizes heterogeneously.

At the same time, the present analysis does not establish a causal pathway from any measured geometric statistic to an accessibility endpoint. The intervention parameter α is the causal treatment: changing the message-passing operator and retraining the model causally changes the learned representation and the measured geometric and accessibility quantities. By contrast, the relations

$$\text{centroid-orthogonal contraction} \;\longrightarrow\; R_P$$

and

$$\text{centroid reconfiguration} \;\longrightarrow\; A_P \text{ or } D_L$$

are not identified by the present experiments. The radial and centroid measurements should therefore be interpreted as post-intervention explanatory correlates that make the observed dose response geometrically intelligible, rather than as estimated mediators.

The resulting picture is more specific than "less degree dependence implies better geometry." Target-degree normalization causally changes several geometric properties of the retrained representation, but these properties do not move in lockstep. Moderate normalization is accompanied by a strong centroid-orthogonal contraction and improved prototype recovery before major centroid degradation emerges. Under stronger normalization, the orthogonal component contracts further, the full-radius degree ordering collapses and eventually reverses, and class-selective centroid reconfiguration becomes more pronounced while supervised accessibility declines. These co-occurring responses motivate the explanatory account in the main text without identifying a unique mediation pathway.

AI Disclosure

We used AI tools only for tasks with recommended disclosure. No generative AI was used for tasks with required disclosure, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proofs,

assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing research methods, assisting with translation, cleaning or reformatting datasets, supporting qualitative or thematic data analysis, or interpreting results.

We used the web version of DeepSeek v4 Flash (accessed September 2026) and ChatGPT 5.6Sol (accessed September 2026) to polish and improve the readability of most paragraphs of the manuscript. It was used only for language editing and did not generate research ideas, methods, results, or interpretations. All AI-polished text was reviewed and edited by the authors, who verified that the meaning, technical claims, and terminology were preserved.

We used Codex (accessed September 2026) to generate code for the four figures in the paper. The generated code was executed to produce the figures, and the authors verified the resulting figures for correctness and consistency with the underlying data and experiments.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, figures, references, and artifacts produced with the aid of generative AI. For transparency, the removed references were Kornblith 2019, Ren 2018, Networked Inequality, and Classic GNNs; the retained and repositioned works were Wang 2022 structural fairness, GraphPatcher, DropEdge, DGN, and GRANOLA.