THE POST-GCN DECADE REVISITED: CURVATURE-STRATIFIED EVALUATION OF RELATIONAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Current evaluation practices in relational learning rely heavily on flat leaderboards that average performance across heterogeneous datasets, implicitly assuming a uniform underlying structure. We show that this assumption introduces systematic bias: it obscures geometry-dependent performance variations and can lead to misleading conclusions about model generalization. In this work, we identify intrinsic geometry as a key latent factor governing model effectiveness. We demonstrate that conventional aggregated metrics mask critical performance trade-offs that only become visible when datasets are stratified by their geometric properties. To address this issue, we introduce a curvature-stratified evaluation framework that partitions datasets into positive, negative, and near-zero curvature regimes. Our benchmark evaluates 18 representative models—including Graph Convolutional Networks (GCNs), Graph Foundation Models (GFMs), and tabular learning methods—across 14 datasets. We find that model rankings are highly stable within each curvature regime but shift significantly across regimes, indicating that performance is fundamentally geometry-dependent rather than universally transferable. Notably, we identify regimes where GFMs offer diminishing returns compared to geometry-aligned GNNs. Based on these findings, we propose a geometry-aware evaluation protocol that yields more reliable and interpretable comparisons than standard aggregated benchmarks. We release all code, curvature-stratified dataset splits, and evaluation tools to support reproducible and rigorous assessment of future relational learning methods.

1 INTRODUCTION

Relational learning seeks to model data where instances are interconnected through explicit or implicit dependencies (Fey et al., 2023; Robinson et al., 2024). This structure is ubiquitous, spanning citation networks and molecular graphs to tabular schemas linked by foreign-key joins (Morris et al., 2020; Wu et al., 2018; Kim et al., 2024). Despite their disparate origins, these settings share a fundamental abstraction: relations induce paths and higher-order connectivity that define the data's underlying topology (Bronstein et al., 2021; Ollivier, 2009). This perspective allows us to treat both graph-structured and table-derived data as instances of a common finite-metric space, amenable to unified relational analysis (Robinson et al., 2024).

Over the past decade, Graph Convolutional Networks (GCNs) have emerged as the dominant paradigm, leveraging message-passing to capture neighborhood dependencies (Kipf & Welling, 2017; Hamilton et al., 2017; Veličković et al., 2018; Schlichtkrull et al., 2018). More recently, the field has moved toward high-capacity Graph Foundation Models (GFMs) (Wang et al., 2025c; Zhao et al., 2024; Yu et al., 2024; Wang et al., 2025b) and non-Euclidean architectures (Nickel & Kiela, 2017; Chami et al., 2019). This rapid expansion has been fueled by large-scale benchmarks that provide a standardized arena for competition (You et al., 2020). However, these benchmarks almost exclusively rely on "flat" leaderboards: performance is averaged across heterogeneous datasets, implicitly assuming that "relational data" is a monolithic category with a uniform underlying structure (Wu et al., 2018; Hu et al., 2020; Dwivedi et al., 2023).

We argue that this assumption is fundamentally flawed and has led to a systematic evaluation bias. Relational datasets differ not only in scale or sparsity but in their intrinsic geometry—the latent curvature that governs how information propagates and how representations collapse (Ollivier, 2009;

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.

Forman, 2003; Ni et al., 2015; Chami et al., 2019). By aggregating results across geometrically distinct datasets (e.g., merging tree-like citation networks with grid-like molecules), current evaluation protocols marginalize over structural differences (Wu et al., 2018; Hu et al., 2020). This obscures systematic performance trade-offs, making it impossible to discern whether a model's success is due to architectural superiority or a lucky alignment with the dominant geometry of the benchmark's dataset mix (Dwivedi et al., 2023; You et al., 2020).

Graph curvature provides more than a geometric description of relational data: discrete curvature has been shown to capture structural properties such as local clustering, community organization, information-routing efficiency, and graph bottlenecks (Tian et al., 2026; Zhou & Jiao, 2026). These properties are directly relevant to graph learning, where curvature has been connected to over-smoothing and over-squashing, and has motivated curvature-guided rewiring, aggregation, and geometry-aware architectures (Topping et al., 2022; Black et al., 2023; Caich & Abbahaddou, 2026). Recent work further suggests that curvature can provide a unified geometric lens on information propagation and generalization in GNNs (Caich & Abbahaddou, 2026). Accordingly, a growing body of work has introduced hyperbolic, spherical, mixed-curvature, and adaptive models to match different graph geometries (Nickel & Kiela, 2017; Ganea et al., 2018; Liu et al., 2019; Skopek et al., 2020; Xiong et al., 2022; Grover et al., 2025). Yet evaluation has remained largely geometry-agnostic: specialized models are still commonly compared through global averages over structurally heterogeneous datasets. This creates a mismatch between model design and model assessment: we design for geometry, yet evaluate for the "average".

In this work, we show that intrinsic geometry helps explain model effectiveness: aggregate metrics conceal ranking shifts across geometric regimes, even when rankings are stable within a regime. We also identify settings where emerging GFMs offer limited gains over simpler, geometry-aligned GNNs. To reveal these patterns, we introduce CURVBENCH, a curvature-stratified audit of 14 datasets and 18 models. It groups natural graphs into positive, negative, and near-zero curvature regimes and examines table-derived graphs separately, replacing a single global leaderboard with regime-conditioned diagnostics.

Our contributions are as follows:

- We develop a framework based on curvature statistics (mean and skewness) to categorize both graph and tabular datasets into geometrically meaningful regimes.
- We evaluate 18 methods—spanning GCNs, GFMs, and tabular models—revealing that performance is fundamentally geometry-dependent rather than universally transferable.
- We formalize a regime-stratified methodology that replaces flat leaderboards with geometry-aware comparisons, providing more reliable and interpretable insights for model selection.
- We release all code, curvature-stratified splits, and diagnostic tools to support a more rigorous and structurally-aware assessment of future relational learning research.

2 GRAPH CURVATURE

Graphs as finite metric spaces. Let $G = (V, E, \mathbf{X})$ be an attributed graph with adjacency matrix $\mathbf{A}$. We regard G as a finite metric space (V, d_G), where $d_G(u, v)$ is the shortest-path distance between nodes u and v. For disconnected graphs, all quantities are computed over finite-distance pairs. Let

$$\mathcal{N}(m) = \{v \in V : (m, v) \in E\}, \qquad \mathrm{diam}_f(G) = \max_{u,v:\, d_G(u,v)<\infty} d_G(u, v).$$

Midpoint curvature residual. For a center node m, we approximate a local metric section by an unordered neighbor pair $\{b, c\} \subset \mathcal{N}(m)$, and use an anchor node $a \neq m$ to probe deviation from Euclidean midpoint geometry (Burago et al., 2001; Bridson & Haefliger, 1999). For every valid quadruple $(a, b, c; m)$, we define

$$\Delta_G(a, b, c; m) = d_G(a, m)^2 + \frac{1}{4} d_G(b, c)^2 - \frac{1}{2}\left(d_G(a, b)^2 + d_G(a, c)^2\right),$$

and the normalized midpoint curvature residual

$$\xi_G(a, b, c; m) = \frac{\Delta_G(a, b, c; m)}{2 d_G(a, m)}. \tag{1}$$

The numerator is zero in a Euclidean space when m is the actual midpoint of b and c (the parallelogram identity). Here b, c are graph neighbors of m, which need not make m an exact midpoint; triangles and the shortest-path metric can affect the residual even on grid-like graphs. We therefore use its sign as an *operational metric signature*, rather than identifying it with sectional, Forman, or Ollivier–Ricci curvature. Its relation to over-squashing is a motivation, not a measured equivalence (Topping et al., 2022; Black et al., 2023). The stated analysis uses unit-hop shortest-path distances; a weighted-graph extension would require specifying a weighted geodesic and an appropriate midpoint sampling rule.

Node-level curvature. Let

$$\mathcal{P}_m = \big\{\{b,c\} : b,c \in \mathcal{N}(m),\, b < c\big\}, \qquad \mathcal{A}_m = \{a \in V \setminus \{m\} : 0 < d_G(a,m) < \infty\}.$$

The raw curvature estimate of node m is

$$\widehat{\kappa}_G(m) = \frac{1}{|\mathcal{P}_m|} \sum_{\{b,c\}\in\mathcal{P}_m} \frac{1}{|\mathcal{A}_m|} \sum_{a\in\mathcal{A}_m} \xi_G(a,b,c;m), \tag{2}$$

with $\widehat{\kappa}_G(m) = 0$ when $|\mathcal{P}_m| = 0$. For cross-dataset comparison, we use the diameter-normalized relative curvature

$$\kappa_G(m) = \frac{\widehat{\kappa}_G(m)}{\mathrm{diam}_f(G)}. \tag{3}$$

Graph-level curvature distribution. CURVBENCH represents each graph by the empirical distribution of its node-level relative curvature values $\mu_G = \frac{1}{|V|}\sum_{m\in V} \theta_{\kappa_G(m)}$, where θ_x denotes the Dirac measure, i.e., a unit point mass at value x. Thus, μ_G is the empirical distribution of $\{\kappa_G(m) : m \in V\}$. We summarize μ_G by its mean curvature $\bar{\kappa}(G) = \frac{1}{|V|}\sum_{m\in V} \kappa_G(m)$ and by its curvature skewness. Let

$$\sigma_\kappa(G) = \left(\frac{1}{|V|}\sum_{m\in V} \big(\kappa_G(m) - \bar{\kappa}(G)\big)^2\right)^{1/2}. \tag{4}$$

The curvature skewness is the third standardized central moment (Joanes & Gill, 1998)

$$\gamma_\kappa(G) = \begin{cases} \dfrac{1}{|V|}\displaystyle\sum_{m\in V} \left(\frac{\kappa_G(m) - \bar{\kappa}(G)}{\sigma_\kappa(G)}\right)^3, & \sigma_\kappa(G) > 0, \\ 0, & \sigma_\kappa(G) = 0. \end{cases} \tag{5}$$

While $\bar{\kappa}(G)$ quantifies the average signed curvature, $\gamma_\kappa(G)$ captures the asymmetry of the curvature distribution. Specifically, positive values denote a right-skewed tail of high-curvature nodes, negative values indicate a left-skewed tail of low-curvature nodes, and values near zero suggest a balanced distribution. Detailed code and descriptions for metrics are given in Appendix E.

3 The Setup of CurvBench

3.1 Datasets

CURVBENCH evaluates 14 relational datasets, including nine natural graphs and five table-derived graphs, spanning diverse scales, homophily levels, and application domains. Geometric descriptors—mean curvature $\bar{\kappa}(G)$ and curvature skewness $\gamma_\kappa(G)$—characterize the underlying structural regimes. Based on these metrics, we partition the natural graphs in Table 1(a) into three regimes: (i) *near-zero geometry*, where $|\bar{\kappa}(G)| < 0.01$ and $|\gamma_\kappa(G)| < 0.5$; among the remaining graphs, (ii) *positively curved*, where $\bar{\kappa}(G) > 0$, and (iii) *negatively curved*, where $\bar{\kappa}(G) < 0$. The skewness criterion prevents a near-zero mean from masking strongly asymmetric curvature distributions. Table-derived graphs in Table 1(b) exhibit a distinct pattern: four medical datasets have nearly zero mean curvature, $\bar{\kappa}(G) \in [0.00021, 0.00034]$, but strongly positive skewness, $\gamma_\kappa(G) \in [4.21, 12.07]$, indicating pronounced positive-curvature tails despite an apparently flat average profile. We therefore analyze table-derived graphs separately rather than forcing these schema-induced, strongly skewed profiles into the natural-graph sign regimes.

Table 1: Statistics of the datasets in CURVBENCH.

(a) Natural graph datasets

Regime	Dataset	Domain	#Nodes	#Edges	Homophily	Avg. Degree	#Features	#Classes	Mean Curv. $\bar{\kappa}(G)$	Skewness $\gamma_\kappa(G)$
Near-zero	Cora Yang et al. (2022)	Citation	2,708	5,278	0.8100	3.90	1,433	7	0.00749	0.08401
	Citeseer Yang et al. (2022)	Citation	3,327	4,552	0.7355	2.74	3,703	6	0.00222	0.38363
	PubMed Yang et al. (2022)	Citation	19,717	44,324	0.8024	4.50	500	3	0.00678	0.43122
Positive	Cornell Pei et al. (2020)	Webpage/WebKB	183	298	0.1309	1.63	1,703	5	0.01050	0.81561
	Airport (Chami et al., 2019)	Transportation	7,543	18,508	0.4289	4.91	7,543	4	0.00213	1.33127
	Actor Pei et al. (2020)	Wikipedia	7,600	30,019	0.2188	3.95	932	5	0.12039	1.30001
Negative	Disease (Chami et al., 2019)	Epidemiological	1,044	1,042	0.8752	0.998	1,000	2	-0.00335	-1.48057
	Telecom Zhou et al. (2022)	Telecommunication	41,143	41,424	0.5620	1.01	240	3	-1.14371	-11.82744
	CS_Phds Rossi & Ahmed (2015)	Academic/Social	1,025	1,043	0.2819	2.04	16	4	-0.00301	-1.53958

(b) Table-derived datasets

Dataset	Domain	#Orig. Tables	#Rows	#Columns	#Nodes	#Edges	Avg. Degree	#Features	#Classes	Mean Curv. $\bar{\kappa}(G)$	Skewness $\gamma_\kappa(G)$
Carcinogenesis (Muggleton, 1997)	Medicine	6	27,570	23	28,027	8,982	0.6410	300	3	0.00034	9.42658
Hepatitis	Medicine	7	12,927	26	12,927	13,016	2.0138	300	3	0.00024	4.21239
PTE Mutlu & Karagoz (2016)	Medicine	38	29,762	76	29,850	18,805	1.2600	300	3	0.00031	9.74080
Toxicology	Medicine	4	49,239	11	49,813	18,267	0.7334	300	3	0.00021	12.06911
F1 Robinson et al. (2024)	Sports	9	97,606	77	97,606	192,560	3.9457	300	40	1.11301	-2.26907

3.2 Compared models

We evaluate 18 representative models grouped by their geometric inductive biases, enabling performance to be interpreted through curvature-conditioned behavior rather than a single aggregate leaderboard.

Geometry-agnostic and Euclidean methods. This group includes MLP as a feature-only baseline, along with GCN, GAT, and GraphSAGE as standard flat-space message passing architectures, and PCNet for Euclidean spectral filtering (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017; Li et al., 2024). These models serve as a reference regime, capturing scenarios where flat aggregation and feature separability are sufficient, particularly in near-zero-curvature settings.

Hyperbolic methods. HGNN, HGCN, HAT, and HyboNet embed graph representations in negatively curved spaces via Riemannian message passing, hyperboloid convolution, or hyperbolic attention (Liu et al., 2019; Chami et al., 2019; Zhang et al., 2021a; Chen et al., 2022). They test whether fixed negative-curvature inductive biases are advantageous for hierarchical or tree-like structures.

Flexible-geometry methods. QGCN and CUSP relax the assumption of a single global geometry by employing pseudo-Riemannian metrics or mixed-curvature spectral filtering (Xiong et al., 2022; Grover et al., 2025). These models are designed for graphs with heterogeneous local structure, where multiple geometric regimes may coexist.

Adaptive Riemannian method. GraphMoRE dynamically assigns nodes to geometry-specific experts through topology-aware gating, constructing personalized mixed-curvature representations (Guo et al., 2025b). This setting tests whether local curvature heterogeneity is better handled through adaptive routing rather than fixed global geometry.

Graph foundation models (GFMs). We include recent GFMs to examine whether multi-domain pretraining mitigates or preserves curvature-dependent behavior. GCOPE (Zhao et al., 2024), MDGPT (Yu et al., 2024), and SAMGPT (Yu et al., 2025) rely on token-based or prompt-based adaptation; MDGFM (Wang et al., 2025b) incorporates structure learning; GraphGluing (Sun et al., 2026a) performs manifold-level alignment; and SA2GFM (Shi et al., 2026) injects hierarchical structural semantics.

Together, this suite enables a systematic comparison between classical geometric models and modern pretrain-and-adapt approaches under a unified curvature-stratified evaluation protocol.

3.3 Implementation details

Dataset settings for GFMs. For GFMs, we use the reported three-fold *dataset-level* protocol: each fold pretrains on six natural graphs (two per regime) and evaluates on three held-out graphs (one per regime), with 1- and 5-shot target supervision. The graph-model and GFM results share target datasets but differ in pretraining and label budgets, so cross-family scores describe the settings tested rather than an equal-supervision comparison. Within each graph, the supplied node splits were not selected to match node-level curvature distributions; consequently, we do not claim a node-level curvature generalization test. The fold assignments and label processing are detailed in Appendix B.

Hyperparameter optimization. We adopt default hyperparameters from the original implementations whenever available and search predefined ranges for key hyperparameters of each method. For each dataset–task pair, we select configurations by validation performance. The reported results average five random seeds and include a mean and standard deviation. Further details appear in Appendix C.

4 Motivation and Theoretical Grounding

CURVBENCH is motivated by two fundamental questions. First, why should graph curvature influence the suitability of a representation geometry? Second, if curvature systematically affects model behavior, how should models be compared across heterogeneous regimes?

Standard evaluation protocols implicitly assume that a single aggregate leaderboard provides a meaningful comparison. Formally, a model M is evaluated under a benchmark mixture as

$$R_{\pi,t}(M) = \sum_{r \in \mathcal{R}} \pi_r \, R_{r,t}(M),$$

where r indexes curvature regimes in $\mathcal{R}$, t denotes the task, and π_r are mixture weights over regimes. $R_{r,t}(M)$ denotes the average evaluation score of model M on task t within curvature regime r.

However, this aggregation obscures regime-dependent behavior. If two models exchange their relative performance across regimes, then varying the mixture weights π can reverse their global ranking while leaving all per-dataset results unchanged. Consequently, a global leaderboard is not an intrinsic property of a model, but rather a function of the benchmark composition.

This observation motivates a shift from global rankings to *regime-conditioned evaluation*. CURVBENCH preserves two key structures: (i) curvature mismatch as a fundamental limitation on representation geometry, and (ii) regime-conditioned performance gaps as indicators of preference variation across geometric settings. All proofs in this section are in Appendix D.

4.1 Curvature mismatch as a metric obstruction

Let (V, d_G) be the finite metric space induced by a graph G, and let $h : V \to \mathcal{M}$ be an embedding into a representation metric space $(\mathcal{M}, d_{\mathcal{M}})$. A quadruple $q = (a, b, c; m)$ is valid if $b \neq c$, $b, c \in \mathcal{N}(m)$, and $0 < d_G(a, m) < \infty$. For any metric space $(\mathcal{X}, d_{\mathcal{X}})$ whose underlying points contain a, b, c, m, we define

$$\xi_{\mathcal{X}}(q) = \frac{d_{\mathcal{X}}(a,m)^2 + \frac{1}{4} d_{\mathcal{X}}(b,c)^2 - \frac{1}{2}\left(d_{\mathcal{X}}(a,b)^2 + d_{\mathcal{X}}(a,c)^2\right)}{2 d_{\mathcal{X}}(a,m)}, \tag{6}$$

where $\xi_{\mathcal{X}}(q)$ is the midpoint curvature residual used by CURVBENCH, written for an arbitrary metric space. Since $\xi_{\mathcal{X}}$ is homogeneous of degree one in distances, we compare graph and representation metrics after a global scale calibration (Bourgain, 1985; Linial et al., 1995). For $\lambda > 0$, define

$$\mathrm{dis}_{\infty}^{\lambda}(h; G, \mathcal{M}) = \max_{u,v:\, d_G(u,v) < \infty} \left| d_{\mathcal{M}}(h(u), h(v)) - \lambda d_G(u, v) \right|. \tag{7}$$

For any valid quadruple $q = (a, b, c; m)$, we denote by $h(q) = (h(a), h(b), h(c); h(m))$ the corresponding quadruple after embedding.

Theorem 4.1 (Curvature mismatch lower-bounds metric distortion)**.** *Let $D = \mathrm{diam}_f(G)$ and $\delta = \mathrm{dis}_{\infty}^{\lambda}(h; G, \mathcal{M})$. If $\delta < \lambda/2$, then for every valid quadruple q in G,*

$$|\xi_{\mathcal{M}}(h(q)) - \lambda \xi_G(q)| \leq C_D \delta, \tag{8}$$

where one admissible constant is $C_D = \frac{1}{2} + \frac{5}{8}(2D+1)^2 + \frac{5}{4}(2D+1)$.

Consequently, for any distribution $\mathcal{Q}_G$ over valid quadruples,

$$\mathbb{E}_{q \sim \mathcal{Q}_G}\left[|\xi_{\mathcal{M}}(h(q)) - \lambda \xi_G(q)| \right] \leq C_D \, \mathrm{dis}_{\infty}^{\lambda}(h; G, \mathcal{M}). \tag{9}$$

For an embedding class $\mathcal{H}$, define $\mathrm{Dist}_{\mathcal{H}}^{\lambda}(G, \mathcal{M}) = \inf_{h \in \mathcal{H}} \mathrm{dis}_{\infty}^{\lambda}(h; G, \mathcal{M})$, and

$$\eta_{\mathcal{H}}^{\lambda}(G, \mathcal{M}) = \inf_{h \in \mathcal{H}} \mathbb{E}_{q \sim \mathcal{Q}_G}\left[|\xi_{\mathcal{M}}(h(q)) - \lambda \xi_G(q)| \right]. \tag{10}$$

If $\mathrm{Dist}^{\lambda}_{\mathcal{H}}(G, \mathcal{M}) < \lambda/2$*, then*

$$\mathrm{Dist}^{\lambda}_{\mathcal{H}}(G, \mathcal{M}) \geq \frac{\eta^{\lambda}_{\mathcal{H}}(G, \mathcal{M})}{C_D}. \tag{11}$$

The result follows from the fact that ξ is Lipschitz with respect to the four distances $d(a, m)$, $d(b, c)$, $d(a, b)$, and $d(a, c)$, provided the denominator is bounded away from zero; the condition $\delta < \lambda/2$ ensures this for all valid quadruples.

This establishes a direct link between metric distortion and curvature preservation: any embedding that approximately preserves pairwise distances must also preserve midpoint curvature residuals up to controlled error. Conversely, systematic mismatch in curvature residuals certifies the presence of non-negligible metric distortion.

This offers a geometric mechanism for regime-dependent behavior: representations whose induced metric better matches the graph's residual structure can incur lower distortion, while mixed or adaptive geometries can accommodate greater local heterogeneity.

4.2 Regime-conditioned orders estimate preference variation

The previous result explains why different curvature regimes may favor different representation geometries. We now show what CURVBENCH measures when it compares model orders within and across regimes.

Let $\mathcal{M}_0 = \{M_1, \ldots, M_N\}$ be the evaluated models. For a graph G and task t, let $\ell_i(G)$ be the empirical loss of model M_i. For tolerance $\epsilon \geq 0$, define the pairwise comparison state

$$Z^{\epsilon}_{ij}(G) = \mathrm{sign}_{\epsilon}\big(\ell_j(G) - \ell_i(G)\big), \qquad \mathrm{sign}_{\epsilon}(x) = \mathrm{sign}(x)\mathbf{1}\{|x| \geq \epsilon\}. \tag{12}$$

The state 0 treats statistically indistinguishable models as incomparable, so the induced order is partial rather than artificially total. For each regime $r \in \mathcal{R}$, define

$$p^{z}_{ij,r} = \mathbb{P}_{G \sim \mathcal{D}_{r,t}}\left(Z^{\epsilon}_{ij}(G) = z\right), \qquad z \in \{-1, 0, +1\}. \tag{13}$$

The normalized distance between two empirical partial orders is

$$d_{\epsilon}(G, G') = \frac{1}{\binom{N}{2}} \sum_{i<j} \mathbf{1}\left\{Z^{\epsilon}_{ij}(G) \neq Z^{\epsilon}_{ij}(G')\right\}. \tag{14}$$

Theorem 4.2 (Within–cross order gap is between-regime preference variance)**.** *Assume $K = |\mathcal{R}| \geq 2$ regimes are sampled uniformly. Let G, G' be independent graphs. Under within-regime sampling, both graphs are drawn from the same randomly selected regime. Under cross-regime sampling, they are drawn from two different randomly selected regimes. Then*

$$\begin{aligned} &\mathbb{E}\left[d_{\epsilon}(G, G') \mid \text{cross}\right] - \mathbb{E}\left[d_{\epsilon}(G, G') \mid \text{within}\right] \\ &\quad = \frac{K}{(K-1)\binom{N}{2}} \sum_{i<j} \sum_{z \in \{-1,0,+1\}} \mathrm{Var}_{r \sim \mathrm{Unif}(\mathcal{R})}\left(p^{z}_{ij,r}\right) \geq 0. \end{aligned} \tag{15}$$

The gap is zero if and only if every model-pair comparison distribution $\{p^{z}_{ij,r}\}_z$ is invariant across curvature regimes.

For a fixed model pair (i, j), within-regime agreement depends on $\sum_z (p^{z}_{ij,r})^2$, while cross-regime depends on $\sum_z p^{z}_{ij,r} p^{z}_{ij,s}$ for $r \neq s$. The difference between these quantities is exactly the variance of the comparison-state probabilities across regimes. Averaging over all model pairs yields Eq. (15).

Therefore, the within–cross order gap is not merely an empirical observation; it provides a direct estimate of between-regime variation in model preferences. If curvature has no effect on model behavior, then $p^{z}_{ij,r}$ is invariant across regimes and the gap vanishes. Conversely, if different curvature regimes favor different inductive biases, the variance is strictly positive, and cross-regime rankings are necessarily less stable.

A global leaderboard collapses this variability into a single aggregate score, obscuring regime-dependent behavior. In contrast, regime-conditioned partial orders preserve this information, enabling a more faithful comparison of models across heterogeneous graph geometries.

5 Experimental Results and Analyses

5.1 Curvature-stratified Model Rankings

Rather than relying on a single global leaderboard, we evaluate whether model-induced rankings are more consistent within the same curvature regime than across different regimes. This aligns with the central premise of CURVBENCH: model performance is governed by the interaction between a model's geometric inductive bias and the intrinsic geometry of the graph.

Table 2: Performance on natural graph datasets: (a) node classification accuracy and (b) link prediction AUC. Top-3 results are highlighted with **red bold**, red, and **bold**.

(a) Node classification accuracy

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCN	80.36±0.71	68.68±0.65	78.12±0.28	79.18±0.98	38.37±3.52	31.31±0.62	83.82±5.58	85.85±0.64	35.51±2.87
GAT	80.72±0.70	67.50±1.64	77.08±0.32	82.82±0.78	44.32±4.52	28.67±0.60	**90.62±1.41**	79.73±0.19	26.83±0.00
GraphSAGE	**88.30±0.21**	74.89±0.65	88.48±0.05	48.80±0.27	**73.51±3.52**	32.84±0.56	95.60±1.45	92.90±3.08	26.73±6.15
MLP	56.12±1.05	54.18±0.87	71.27±0.38	85.07±0.55	68.10±2.26	**37.46±0.62**	79.90±0.00	88.15±0.04	26.83±0.00
PCNet	88.08±0.44	**75.59±0.25**	**89.97±0.11**	45.51±0.13	**61.08±4.52**	**33.45±0.97**	78.56±0.92	87.49±0.04	31.51±0.44
HAT	**81.60±0.32**	**70.99±0.28**	**78.74±0.46**	59.22±5.53	36.84±0.03	34.64±0.44	77.51±0.30	87.92±0.02	26.82±0.00
HGNN	78.52±0.63	67.62±0.81	76.54±0.43	**83.51±2.47**	**61.08±1.32**	28.92±0.68	77.72±2.15	**93.16±0.97**	24.41±2.87
HyboNet	75.16±0.84	70.23±1.20	73.58±0.45	60.88±4.17	36.22±1.06	26.67±1.32	77.01±4.59	62.03±7.32	26.73±0.19
HGCNN	76.74±0.78	67.22±1.01	75.88±0.33	60.23±2.20	**61.08±0.96**	28.80±0.23	77.92±1.56	**93.16±1.70**	43.63±2.86
CUSP	76.94±0.95	68.20±1.28	66.36±2.31	58.65±2.24	40.54±1.00	24.81±1.26	85.79±1.87	66.73±5.01	29.65±3.47
QGCN	79.80±0.41	67.32±0.26	75.90±1.03	61.07±0.74	54.59±2.02	26.74±0.55	83.31±1.42	**98.25±0.05**	**45.39±2.33**
GraphMoRE	81.06±0.33	68.30±0.78	76.34±1.12	**90.42±1.32**	40.54±3.42	24.49±0.81	**96.11±0.77**	93.40±0.31	**37.45±2.82**

(b) Link prediction AUC

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCN	**91.87±1.01**	92.09±0.87	92.82±0.31	94.79±0.57	64.60±5.81	80.11±0.65	47.78±4.97	71.53±0.40	42.50±0.98
GAT	92.07±0.42	**92.83±0.62**	91.77±0.20	93.80±0.45	**70.06±8.74**	**80.48±0.88**	49.64±5.47	73.87±0.61	**61.35±4.24**
GraphSAGE	72.74±2.42	74.40±1.73	84.02±1.16	79.04±0.83	58.59±9.80	61.34±1.21	49.95±0.11	60.52±1.05	36.91±0.02
MLP	83.08±0.96	88.88±1.54	86.97±0.56	91.67±0.62	66.06±6.03	69.95±0.67	50.72±4.98	**73.76±0.76**	50.00±0.00
PCNet	74.40±1.94	72.24±1.38	93.39±0.27	79.67±0.78	54.61±9.92	61.69±0.68	59.11±3.97	64.61±1.02	**84.28±1.22**
HGNN	74.33±0.49	86.29±0.38	92.23±0.20	92.81±0.27	66.02±5.47	68.79±0.32	51.87±1.77	**76.23±0.20**	49.72±4.45
HGCNN	83.54±1.72	88.03±0.66	93.83±0.06	93.33±0.56	69.75±0.58	79.37±0.35	**62.59±7.89**	56.01±1.70	52.45±1.46
HyboNet	87.88±2.11	77.40±0.82	91.19±0.60	96.29±0.63	**66.99±3.77**	82.18±1.05	40.56±2.89	49.84±0.61	62.54±4.50
CUSP	88.33±1.56	93.98±1.40	63.74±0.70	75.74±0.98	64.67±7.20	71.93±1.11	27.20±5.78	68.39±0.75	40.34±5.84
QGCN	88.00±0.42	87.24±0.59	**93.51±0.20**	**95.92±0.14**	62.58±3.66	79.19±0.41	72.74±1.53	65.64±0.67	54.29±1.50
GraphMoRE	**96.50±0.36**	**98.39±0.18**	**98.63±0.20**	**96.30±0.42**	64.19±4.81	**85.54±0.45**	**74.22±2.30**	72.92±0.44	42.10±7.47

Observation 1: Curvature regimes induce coherent top-model partial orders. We first test whether curvature regimes organize model behavior by comparing dataset-induced model rankings. For each dataset, models are sorted by mean performance and converted into a top-3 truncated ranking: the top three models retain their exact ranks, while all remaining models are grouped into a shared lower tier. This focuses the analysis on the dominant inductive biases revealed by the benchmark.

On node classification, the proposed curvature grouping yields substantially higher within-regime than cross-regime ranking consistency. The average top-3 truncated Spearman correlation (Spearman, 1904) is 0.539 within regimes versus 0.036 across regimes, corresponding to a gap of 0.503. The same trend holds for Kendall correlation (0.512 vs. 0.031) (Kendall, 1938) and top-3 Jaccard overlap (0.567 vs. 0.159) (Jaccard, 1901).

Importantly, among all 280 balanced 3-3-3 partitions of the nine datasets, the proposed curvature grouping achieves the **largest within-minus-cross gap** under all three metrics, yielding an exact regrouping significance of $p = 1/280 = 0.0036$.

Results in Table 2 provide strong evidence that curvature is not merely a post-hoc descriptor, but a meaningful organizing principle for model behavior. Datasets within the same curvature regime exhibit substantially more similar top-model preferences than datasets across regimes, supporting the use of curvature-conditioned partial orders rather than a single global leaderboard.

Observation 2: Curvature regimes reorganize effective inductive biases. We next examine which model families dominate top-performing positions across regimes. Models are grouped by their geometric inductive biases: Euclidean, fixed hyperbolic, mixed or pseudo-Riemannian, and adaptive Riemannian. The composition of top-performing models changes markedly with curvature.

In the near-zero regime, Euclidean methods dominate, winning all three datasets and occupying 66.7% of the top-3 positions. Moreover, Cora, Citeseer, and PubMed share an almost identical top-3 set—GraphSAGE, PCNet, and HAT—up to minor rank variations, indicating that flat aggregation and spectral filtering are well aligned with near-zero curvature structure.

In the positive regime, Euclidean models remain competitive, winning two of three datasets and accounting for 55.6% of top-3 placements. However, the inductive bias shifts: MLP appears in the top-3 across all datasets, suggesting that in positively curved (compact or clustered) graphs, node attributes become more predictive than relational structure.

The negative regime exhibits the most pronounced reorganization. Mixed and adaptive Riemannian methods win all three datasets and occupy 55.6% of top-3 positions, while Euclidean methods drop to 0%. This sharp transition indicates that Euclidean models are not inherently weak, but rather regime-limited: their failure in negatively curved graphs reflects a mismatch between flat geometry and hierarchical or tree-like structure.

We further quantify this interaction via variance decomposition after per-dataset normalization. Incorporating a family-by-regime interaction term increases explained variance from 0.0596 to 0.1063, with the interaction accounting for 43.95% of the explained variance. Results show that model performance is governed by the alignment between geometric inductive bias and graph curvature, rather than by a globally dominant model family.

5.2 Geometry-conditioned behavior of Graph Foundation Models

Table 3: Performance of GFMs under 1-shot and 5-shot scenarios. OOM means out-of-memory.

1-shot scenario									
Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCOPE	33.19±6.05	**37.38±7.46**	41.49±4.35	**19.22±8.35**	24.62±9.36	**24.30±1.85**	73.08±12.69	**54.82±13.10**	**26.21±2.11**
MDGPT	44.58±7.83	39.04±10.53	**53.36±10.72**	18.28±17.07	29.26±6.27	20.01±4.33	52.42±9.43	**36.56±12.55**	25.29±2.30
MDGFM	**43.27±7.28**	**41.20±6.31**	51.52±9.34	18.70±5.03	**35.14±9.02**	**20.74±2.15**	57.84±10.77	OOM	25.56±2.11
SAMGPT	**44.64±14.94**	36.03±8.41	45.24±8.45	19.12±9.20	33.84±8.54	19.72±5.88	**60.28±11.04**	45.12±13.49	25.36±6.92
GraphGluing	32.22±1.33	28.48±6.59	**45.90±4.70**	**41.37±2.77**	**32.51±11.25**	24.10±2.25	**79.67±0.18**	OOM	26.15±2.45
SA2GFM	40.25±8.05	29.98±7.81	45.79±8.90	25.63±5.95	20.99±5.45	18.53±2.09	51.12±13.39	OOM	**25.92±2.75**
5-shot scenario									
GCOPE	61.40±1.88	52.42±5.26	58.56±1.79	20.95±4.32	**68.03±4.33**	**24.55±2.07**	79.44±0.58	**72.16±8.37**	26.70±1.95
MDGPT	**60.86±4.86**	**58.68±6.93**	**59.86±6.83**	**22.78±10.44**	44.98±7.18	**21.28±4.21**	54.68±9.71	**38.74±9.13**	26.86±2.27
MDGFM	**64.93±4.43**	58.10±4.55	65.65±5.30	19.92±3.88	60.10±7.78	21.12±1.67	63.55±8.69	OOM	**26.81±2.42**
SAMGPT	64.62±9.89	**53.76±5.70**	56.16±7.27	21.28±6.74	**52.24±6.18**	19.92±6.24	**68.32±9.88**	58.56±11.62	**27.12±6.40**
GraphGluing	52.52±6.06	44.05±2.08	**66.14±1.71**	**42.46±1.11**	40.33±10.72	23.47±1.75	**80.42±0.73**	OOM	26.63±1.67
SA2GFM	50.91±6.57	38.25±4.18	53.40±8.94	25.95±9.77	22.83±7.34	19.35±2.96	56.77±10.84	OOM	26.05±1.87

We further evaluate recent GFMs under the same curvature-stratified protocol. The evaluated methods span representative transfer mechanisms, including virtual-node coordination, domain/structure tokenization, topology alignment, Riemannian gluing, and structure-aware augmentation. This setup enables a sharper question: do GFMs eliminate the need for curvature-aware evaluation, or merely introduce new geometry-conditioned inductive biases?

Observation 3: 1-shot GFM rankings remain curvature-conditioned. The 1-shot regime is most diagnostic, as performance is dominated by transferred inductive bias rather than target supervision. Under top-3 truncated rankings, model orderings are substantially more consistent within curvature regimes than across them: Spearman correlation increases from -0.102 to 0.269, and Kendall correlation from -0.086 to 0.222. Among all 280 balanced 3-3-3 partitions of the nine datasets, our curvature grouping ranks in the top 5% (Spearman) and 6.1% (Kendall), confirming that it captures a meaningful axis of transfer behavior.

Crucially, the leading GFM varies by regime. On near-zero graphs, MDGPT and MDGFM achieve the best 1-shot means (45.66 and 45.33), indicating that token- or prompt-based alignment is well-suited to citation-like structures. On positive graphs, GraphGluing becomes dominant (mean 32.66), largely driven by Airport. On negative graphs, GraphGluing again achieves the strongest available-case performance, while GCOPE is the best complete-coverage method due to OOM failures of several competitors on Telecom.

Overall, GFMs do not collapse CURVBENCH into a geometry-agnostic leaderboard. Instead, performance depends on how inductive biases interact with curvature. A flat 1-shot average favors GraphGluing if OOM cases are ignored, whereas a complete-coverage view favors GCOPE. The

regime-aware view is more informative: MDGFM excels on near-zero graphs, GraphGluing dominates structurally demanding regimes when feasible, and GCOPE offers the most robust coverage.

Observation 4: Extra labels reveal a geometry–scalability frontier. Increasing supervision from 1-shot to 5-shot yields highly uneven gains across regimes. Averaged over GFMs, near-zero graphs improve by 15.95, positive graphs by 6.97, and negative graphs by only 4.49. Dataset-level trends reinforce this pattern: Airport gains 1.84, Actor 0.38, and CS_Phds 0.95. This suggests that citation-like graphs are primarily label-limited, whereas several non-citation graphs remain geometry- or structure-limited even with additional supervision.

This exposes a fundamental trade-off. Geometry-intensive methods such as GraphGluing can achieve strong performance on structurally distinctive regimes, but their advantage is conditional on scalability. In contrast, lighter coordination or prompting approaches—especially GCOPE—are less specialized but provide broader coverage and stronger label elasticity (e.g., GCOPE improves by 14.43 and is among the few methods that handle Telecom).

Therefore, GFM evaluation should jointly report curvature regime, shot elasticity, and feasibility, as OOM behavior is part of practical transfer feasibility rather than a negligible implementation detail.

5.3 Performance on table-derived graphs

Observation 5: Table-derived graphs expose tail-driven geometry. As shown in Table 4, model behavior on table-derived graphs is highly non-uniform. HAT wins three of five datasets and dominates F1 (40.84 vs. 5.04), but drops to the bottom tier on Hepatitis and Toxicology, where Euclidean methods remain strong. This suggests a construction-induced mixed regime: performance depends less on mean curvature $\bar{\kappa}(G)$ and more on asymmetric (tail) curvature mass introduced by table construction. Hyperbolic attention acts as a high-variance specialist, excelling when curvature tails are pronounced, while Euclidean methods provide more stable baselines.

Table 4: Performance on table-derived graphs.

Baselines	Carcinogenesis	Hepatitis	PTE	Toxicology	F1
GCN	57.27±5.07	83.19±0.44	79.66±1.82	**54.78±1.58**	**4.70±0.70**
GAT	60.30±4.59	79.80±1.29	78.33±3.11	52.75±1.29	4.25±0.14
GraphSAGE	65.45±1.27	**81.80±1.30**	81.67±0.00	**55.07±1.02**	4.10±0.14
MLP	54.55±0.00	70.80±1.78	79.00±0.91	**55.07±0.20**	3.96±0.40
PCNet	53.03±2.14	**84.20±1.92**	**81.00±1.49**	52.46±0.65	3.90±0.27
HGNN	62.42±2.42	66.80±0.40	77.00±1.25	53.33±2.13	4.02±0.46
HAT	**70.84±1.47**	59.19±0.44	**85.66±3.02**	40.57±4.09	**40.84±5.77**
HGCNN	61.21±1.21	64.20±0.40	65.33±3.86	51.59±1.48	**4.73±0.16**
HyboNet	43.63±1.76	67.59±5.38	43.33±5.55	44.92±5.55	4.11±0.21
CUSP	57.57±7.66	80.40±1.20	51.66±10.90	54.87±0.57	5.04±0.25
QGCN	**63.33±2.42**	67.20±2.23	55.33±1.25	53.04±2.35	4.47±0.50
GraphMoRE	54.55±5.07	81.00±1.67	78.33±1.05	53.91±0.58	4.16±0.12

Thus, both $\bar{\kappa}(G)$ and $\gamma_\kappa(G)$ are needed to explain performance; a flat average obscures this specialist–robustness trade-off.

Figure 1 further demonstrates that model rankings are highly sensitive to the underlying data regime. A model's standing can shift substantially once evaluation is conditioned on curvature-stratified datasets, indicating that model performance is not an intrinsic, context-independent property.

Additional analyses in Appendix G further show that model comparisons depend jointly on geometry, task, metric choice, label elasticity, coverage, and computational feasibility.

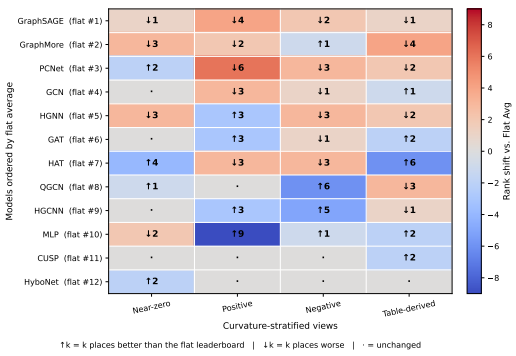


Figure 1: Rank-shift heatmap.

6 Conclusion and Future Work

We introduced CurvBench, a curvature-stratified benchmark for relational learning. Flat leaderboards obscure ranking shifts across geometric regimes, tasks, and metrics, despite greater consistency within regimes. Our findings motivate assessing how model inductive biases interact with data geometry, alongside label elasticity, coverage, specialist–robustness trade-offs, and efficiency. Future evaluations should use stratified, coverage-aware diagnostics to reveal when each model succeeds. Future models could also infer, select, or combine suitable inductive biases from the data. Together, these directions can better characterize the strengths and limitations of graph and foundation models.

AI USE STATEMENT

AI assisted with language editing and organization of existing results. It also assisted with LaTeX formatting and consistency checks. The authors are responsible for checking the claims and results.

REFERENCES

Nurul A Asif, Yeahia Sarker, Ripon K Chakrabortty, Michael J Ryan, Md Hafiz Ahamed, Dip K Saha, Faisal R Badal, Sajal K Das, Md Firoz Ali, Sumaya I Moyeen, et al. Graph neural network: A comprehensive review on non-euclidean space. *Ieee Access*, 9:60588–60606, 2021.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. Understanding oversquashing in gnns through the lens of effective resistance. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2528–2547, 2023. URL https://proceedings.mlr.press/v202/black23a.html.

Jean Bourgain. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52(1-2):46–52, 1985. doi: 10.1007/BF02776078.

Martin R. Bridson and André Haefliger. *Metric Spaces of Non-Positive Curvature*, volume 319 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 1999. ISBN 9783540643241. doi: 10.1007/978-3-662-12494-9. URL https://link.springer.com/book/10.1007/978-3-662-12494-9.

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021. URL https://arxiv.org/abs/2104.13478.

Dmitri Burago, Yuri Burago, and Sergei Ivanov. *A Course in Metric Geometry*, volume 33 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2001. ISBN 9780821821299. URL https://pubs.ams.org/ebooks/gsm/033/.

Rachid Caich and Yassine Abbahaddou. Local-global geometric insights for graph neural networks via entropic curvature. *arXiv preprint arXiv:2607.22381*, 2026.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0415740eaa4d9decbc8da001d3fd805f-Abstract.html.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5672–5686, 2022.

Enjun Du, Siyi Liu, and Yongqi Zhang. Graphoracle: Efficient fully-inductive knowledge graph reasoning via relation-dependency graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19055–19063, 2026.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24 (43):1–48, 2023. URL https://jmlr.org/papers/v24/22-0567.html.

Dmitry Eremeev, Gleb Bazhenov, Oleg Platonov, Artem Babenko, and Liudmila Prokhorenkova. Turning tabular foundation models into graph foundation models. *arXiv preprint arXiv:2508.20906*, 2025.

Matthias Fey, Weihua Hu, Kexin Huang, Jan Eric Lenssen, Rishabh Ranjan, Joshua Robinson, Rex Ying, Jiaxuan You, and Jure Leskovec. Relational deep learning: Graph representation learning on relational databases. *arXiv preprint arXiv:2312.04615*, 2023. URL https://arxiv.org/abs/2312.04615.

Robin Forman. Bochner's method for cell complexes and combinatorial ricci curvature. *Discrete & Computational Geometry*, 29(3):323–374, 2003. doi: 10.1007/s00454-002-0743-x. URL https://link.springer.com/article/10.1007/s00454-002-0743-x.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1805.09112.

Karish Grover, Haiyang Yu, Xiang Song, Qi Zhu, Han Xie, Vassilis N. Ioannidis, and Christos Faloutsos. Spectro-riemannian graph neural networks. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.00401.

Zihao Guo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, Min Zhou, Yisen Gao, and Jianxin Li. Graphmore: Mitigating topological heterogeneity via mixture of riemannian experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11754–11762, 2025a.

Zihao Guo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, Min Zhou, Yisen Gao, and Jianxin Li. Graphmore: Mitigating topological heterogeneity via mixture of riemannian experts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11754–11762, 2025b.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html.

Adrian Hayler, Klemens Flöge, Alan Arazi, Rishabh Ranjan, Jure Leskovec, Felix Birkel, Brendan Roof, Anurag Garg, Kristina Collins, Lydia Sidhoum, et al. Advancing open and reproducible relational learning: Relarena-

\

alpha , $tabpfn - relandrpi$. *arXiv preprint arXiv:2608.16319*, 2026.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et du jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

D. N. Joanes and C. A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998. doi: 10.1111/1467-9884.00122. URL https://academic.oup.com/jrsssd/article-abstract/47/1/183/7123309.

Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. doi: 10.1093/biomet/30.1-2.81.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. Carte: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning*, pp. 23843–23866. PMLR, 2024.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Bingheng Li, Erlin Pan, and Zhao Kang. Pc-conv: Unifying homophily and heterophily with two-fold filtering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 13437–13445, 2024.

Haifeng Li, Jun Cao, Jiawei Zhu, Yu Liu, Qing Zhu, and Guohua Wu. Curvature graph neural network. *Information Sciences*, 592:50–66, 2022.

Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. doi: 10.1007/BF01200757.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/103303dd56a731e377d01f6a37badae3-Abstract.html.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 1150–1160, 2021.

Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020. URL https://arxiv.org/abs/2007.08663.

SH Muggleton. The predictive toxicology evaluation challenge. In *IJCAI-97: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan, August 23-29, 1997*, volume 2, pp. 4. Morgan Kaufmann, 1997.

Alev Mutlu and Pinar Karagoz. Policy-based memoization for ilp-based concept discovery systems. *Journal of Intelligent Information Systems*, 46(1):99–120, 2016.

Chien-Chun Ni, Yu-Yao Lin, Jie Gao, Xianfeng David Gu, and Emil Saucan. Ricci curvature of the internet topology. In *IEEE Conference on Computer Communications*, pp. 2758–2766. IEEE, 2015. URL https://arxiv.org/abs/1501.04138.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/59dfa2df42d9e3d41f5b02bfc32229dd-Abstract.html.

Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256 (3):810–864, 2009. doi: 10.1016/j.jfa.2008.11.001. URL https://arxiv.org/abs/math/0701886.

Hongbin Pei, Bingzhe Wei, Kevin Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *ICLR*, 2020.

Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan Eric Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. Relbench: A benchmark for deep learning on relational databases. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/25cd345233c65fac1fec0ce61d0f7836-Abstract-Datasets_and_Benchmarks_Track.html.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL https://networkrepository.com.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pp. 593–607. Springer, 2018.

Junhua Shi, Qingyun Sun, Haonan Yuan, and Xingcheng Fu. Sa^{2}gfm: Enhancing robust graph foundation models with structure-aware semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15716–15724, 2026. doi: 10.1609/aaai.v40i18.38602. URL https://doi.org/10.1609/aaai.v40i18.38602.

Ondrej Skopek, Octavian-Eugen Ganea, and Gary Bécigneul. Mixed-curvature variational autoencoders. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g6xeSKDS.

Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

Li Sun, Zhenhao Huang, Silei Chen, Lanxu Yang, Junda Ye, Sen Su, and Philip S. Yu. Multi-domain riemannian graph gluing for building graph foundation models. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=G3uNHQpP7J.

Qingyun Sun, Haonan Yuan, Yi Huang, Ziwei Zhang, Xingcheng Fu, Ruijie Wang, Haoyi Zhou, Jia Wu, Jianxin Li, and Philip S Yu. A survey on foundation models for structured data: Tabular, time series, and graphs. 2026b.

Yu Tian, Eleanor Wiesler, and Melanie Weber. Ollivier's ricci curvature on complex-weighted graphs. *arXiv preprint arXiv:2608.10132*, 2026.

Jake Topping, Francesco Di Giovanni, Benjamin P. Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2111.14522.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Shuo Wang, Shunyang Huang, Jinghui Yuan, Zhixiang Shen, and Zhao Kang. Cooperation of experts: Fusing heterogeneous information with large margin. In *International Conference on Machine Learning*, pp. 63169–63185. PMLR, 2025a.

Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. In *International Conference on Machine Learning*, 2025b. URL https://arxiv.org/abs/2502.02017.

Zehong Wang, Zheyuan Liu, Tianyi Ma, Jiazheng Li, Zheyuan Zhang, Xingbo Fu, Yiyang Li, Zhengqing Yuan, Wei Song, Yijun Ma, Qingkai Zeng, Xiusi Chen, Jianan Zhao, Jundong Li, Meng Jiang, Pietro Liò, Nitesh Chawla, Chuxu Zhang, and Yanfang Ye. Graph foundation models: A comprehensive survey. *arXiv preprint arXiv:2505.15116*, 2025c. URL https://arxiv.org/abs/2505.15116.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9(2):513–530, 2018. doi: 10.1039/C7SC02664A. URL https://pubs.rsc.org/en/content/articlelanding/2018/sc/c7sc02664a.

Bo Xiong, Shichao Zhu, Nico Potyka, Shirui Pan, Chuan Zhou, and Steffen Staab. Pseudo-riemannian graph convolutional networks. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/16c628ab12dc4caca8e7712affa6c767-Abstract-Conference.html.

Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv (Cornell University)*, 2022. doi: 10.48550/arxiv.1603.08861.

Junda Ye, Zhongbao Zhang, Li Sun, and Siqiang Luo. Mose: Unveiling structural patterns in graphs via mixture of subgraph experts. *arXiv preprint arXiv:2509.09337*, 2025.

Jiaxuan You, Rex Ying, and Jure Leskovec. Design space for graph neural networks. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c5c3d4fe6b2cc463c7d7ecba17cc9de7-Abstract.html.

Xingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. Text-free multi-domain graph pre-training: Toward graph foundation models. *arXiv preprint arXiv:2405.13934*, 2024. URL https://arxiv.org/abs/2405.13934.

Xingtong Yu, Zechuan Gong, Chang Zhou, Yuan Fang, and Hui Zhang. Samgpt: Text-free graph foundation model for multi-domain pre-training and cross-domain adaptation. In *Proceedings of the ACM Web Conference*, 2025. doi: 10.1145/3696410.3714828. URL https://arxiv.org/abs/2502.05424.

Yiding Zhang, Xiao Wang, Chuan Shi, Xunqiang Jiang, and Yanfang Ye. Hyperbolic graph attention network. *IEEE Transactions on Big Data*, 8(6):1690–1701, 2021a.

Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In *Proceedings of The Web Conference*, pp. 1249–1261, 2021b.

Haihong Zhao, Aochuan Chen, Xiangguo Sun, Hong Cheng, and Jia Li. All in one and one for all: A simple yet effective method towards cross-domain graph pretraining. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. URL https://arxiv.org/abs/2402.09834.

Min Zhou, Bisheng Li, Menglin Yang, and Lujia Pan. Telegraph: A benchmark dataset for hierarchical link prediction. *arXiv (Cornell University)*, abs/2204.07703, 2022. doi: 10.48550/arxiv.2204.07703.

Zexian Zhou and Bo Jiao. Ollivier-ricci curvature in cycle overlap mode. *arXiv preprint arXiv:2606.03317*, 2026.

A BASELINES

A.1 TOPOLOGY-AGNOSTIC AND EUCLIDEAN BASELINES

MLP. We first include a multi-layer perceptron (MLP) as a topology-agnostic baseline. MLP uses node features as input and does not perform message passing over the graph. In the context of our curvature-aware evaluation, MLP plays an important diagnostic role: it measures how much predictive signal can be explained by attributes alone, without using the graph metric induced by edges. If MLP performs competitively on a dataset, the task may be dominated by feature separability rather than geometric structure. Conversely, a substantial gap between MLP and topology-aware models indicates that relational information is necessary. Thus, MLP serves as a curvature-null control that helps separate feature-level effects from geometry-induced inductive bias.

GCN. Graph Convolutional Network (GCN) is one of the most widely used Euclidean message-passing baselines (Kipf & Welling, 2017). It propagates information through a normalized adjacency operator and performs feature smoothing over local neighborhoods. From a geometric perspective, GCN assumes that neighbor aggregation in a flat Euclidean representation space is sufficient to capture the task-relevant graph structure. This makes GCN a natural reference point for evaluating whether non-Euclidean models provide additional benefits beyond standard Euclidean smoothing. Since GCN is also known to behave similarly to a low-pass graph filter, its performance across curvature regimes helps reveal whether flat local aggregation is robust to graphs with positive, negative, or near-zero curvature.

GAT. Graph Attention Network (GAT) extends Euclidean message passing by learning attention weights over neighboring nodes (Veličković et al., 2018). Instead of assigning fixed normalized weights as in GCN, GAT adaptively determines the relative importance of each neighbor during aggregation. This improves the flexibility of local message passing but still operates in Euclidean space. In our benchmark, GAT is used to test whether adaptive neighbor weighting alone can compensate for curvature mismatch. If GAT improves over GCN within a curvature regime, the gain can be attributed to local anisotropic aggregation; if it remains inferior to geometry-aware models, this suggests that attention in a flat space is not sufficient to capture the underlying graph metric.

GraphSAGE. GraphSAGE is an inductive Euclidean GNN that generates node representations by sampling and aggregating features from local neighborhoods (Hamilton et al., 2017). Compared with transductive embedding methods, GraphSAGE learns an aggregation function that can generalize to unseen nodes or graphs. We include GraphSAGE because it represents a widely adopted scalable message-passing paradigm and provides a strong Euclidean baseline for both effectiveness and efficiency. In the curvature-aware setting, GraphSAGE helps evaluate whether inductive neighborhood aggregation can remain stable across different geometric regimes, especially when the local graph metric varies significantly across datasets.

A.2 SPECTRAL AND HOMOPHILY/HETEROPHILY-AWARE EUCLIDEAN MODELS

In addition to classical spatial GNNs, we include spectral or filter-based Euclidean models to distinguish curvature effects from frequency-domain effects. These methods do not explicitly change the representation manifold, but they modify how graph signals are propagated across the spectrum.

PCNet. PCNet instantiates PC-Conv, a two-fold filtering mechanism designed to unify homophilic and heterophilic aggregation (Li et al., 2024). It combines a local low-pass filter with a heterophilic graph heat kernel and uses Poisson-Charlier polynomial filters to capture multi-order information. Although PCNet remains Euclidean, it is important for CURVBENCH because it addresses a different structural axis from curvature: the compatibility between graph topology and label smoothness. Including PCNet allows us to examine whether performance variations attributed to curvature can instead be explained by homophily, heterophily, or spectral filtering flexibility. This distinction is crucial because curvature and homophily are correlated in some datasets but are not equivalent geometric properties.

Hyperbolic models are designed for negatively curved spaces, where volume grows exponentially with radius. Such geometry is well suited to graphs with hierarchical, tree-like, or scale-free structures. These models directly embody the hypothesis that representation geometry should match graph geometry, making them central to our benchmark.

HGNN. Hyperbolic Graph Neural Network (HGNN) generalizes GNN operations to Riemannian manifolds through differentiable exponential and logarithmic maps (Liu et al., 2019). Its message passing is performed by mapping node representations between the hyperbolic manifold and tangent spaces. In CURVBENCH, HGNN serves as an early representative of manifold-based graph learning. It tests whether replacing Euclidean hidden representations with hyperbolic ones improves performance on negatively curved graphs, and whether such benefits persist outside the regime for which hyperbolic geometry is theoretically motivated.

HGCN. Hyperbolic Graph Convolutional Network (HGCN) extends GCN to the hyperboloid model of hyperbolic space (Chami et al., 2019). It maps Euclidean input features into hyperbolic representations, performs neighborhood aggregation through tangent-space operations, and learns layer-wise curvature parameters. HGCN is particularly relevant to our benchmark because its design explicitly targets hierarchical and scale-free graphs. Evaluating HGCN across positive-, negative-, and near-zero-curvature regimes allows us to test whether its reported advantage is tied to negative curvature or whether it generalizes as a broader graph representation mechanism.

HAT. Hyperbolic Graph Attention Network (HAT) transfers the attention mechanism from Euclidean space to hyperbolic geometry (Zhang et al., 2021a). It uses gyrovector operations and hyperbolic proximity to define feature transformation and attention-based aggregation. Compared with HGCN, HAT allows us to isolate the role of attention under a negative-curvature geometry. In our benchmark, HAT helps answer whether hyperbolic attention provides additional benefits over Euclidean attention, and whether such benefits are concentrated in negatively curved graphs.

HyboNet. HyboNet is a fully hyperbolic neural architecture based on the Lorentz model (Chen et al., 2022). Unlike methods that implement most operations in tangent spaces, HyboNet formalizes core neural operations directly in hyperbolic space through Lorentz transformations. This design reduces the reliance on repeated logarithmic and exponential mappings and provides a stronger test of fully hyperbolic inductive bias. In CURVBENCH, HyboNet is included to evaluate whether more faithful hyperbolic computation improves robustness and efficiency across curvature regimes.

A.3 PSEUDO- AND MIXED-CURVATURE MODELS

Real-world graphs often contain multiple geometric patterns simultaneously. A single constant-curvature space may be too restrictive for graphs with both hierarchical and cyclic substructures (Zhang et al., 2021b; Li et al., 2022). We therefore include pseudo- and mixed-curvature models that aim to represent heterogeneous topology beyond purely Euclidean or purely hyperbolic assumptions.

QGCN. Pseudo-Riemannian Graph Convolutional Network (QGCN) extends GCNs to pseudo-Riemannian manifolds with indefinite metrics (Xiong et al., 2022). These manifolds generalize hyperbolic and spherical geometries and provide a flexible representation space for graphs with mixed topologies, such as hierarchical structures with cycles. QGCN is central to our benchmark because it directly challenges the single-curvature assumption: if a graph contains both positive- and negative-curvature regions, a pseudo-Riemannian representation may be more appropriate than a purely hyperbolic or Euclidean one.

CUSP. CUSP is a mixed-curvature spectral GNN that integrates curvature and spectral information (Grover et al., 2025). It introduces a curvature-aware graph Laplacian based on Ollivier-Ricci curvature, applies mixed-curvature graph filters over product manifolds, and uses curvature-based positional encoding for hierarchical pooling. We include CUSP because it explicitly connects discrete graph curvature with spectral filtering, thereby addressing two dimensions of graph structure that are often studied separately. In our benchmark, CUSP helps test whether curvature-aware filtering improves performance in regimes where both geometric and frequency-domain signals are important.

A.4 ADAPTIVE RIEMANNIAN MODELS

Beyond fixed or globally mixed curvature, adaptive Riemannian models attempt to assign different geometric spaces to different nodes or local regions. This is particularly aligned with the motivation of CURVBENCH, where graph datasets may contain construction-induced or locally varying curvature patterns.

GraphMoRE. GraphMoRE introduces a Mixture of Riemannian Experts framework to mitigate topological heterogeneity (Guo et al., 2025b). Instead of embedding all nodes into the same global manifold, it uses a topology-aware gating mechanism to route nodes to different Riemannian experts and constructs personalized mixed-curvature spaces. This model is especially important for our benchmark because it operationalizes the idea that curvature may vary locally within a graph. Its performance across regimes provides evidence for whether adaptive curvature selection is necessary when graphs contain heterogeneous or mixed geometric structures.

A.5 Graph Foundation Models

Graph foundation models (GFMs) have recently emerged as a prominent paradigm in graph learning (Mao et al., 2024). Their central goal is to capture domain-invariant knowledge from graphs across diverse domains and transfer such knowledge to unseen graph domains. Existing GFMs differ substantially in the type of transferable signal they emphasize. Methods such as GCOPE (Zhao et al., 2024) and SAMGPT (Yu et al., 2025) primarily focus on feature-level or prompt-based alignment, whereas approaches such as MDGFM (Wang et al., 2025b) and GraphGluing (Sun et al., 2026a) place greater emphasis on intrinsic topological properties and structure-level transfer. In this sense, different GFMs encode inductive biases from different perspectives. Therefore, rather than categorizing each GFM from a predefined geometric viewpoint, we treat them as representative foundation-model baselines and empirically examine whether their claimed cross-domain transferability truly holds across different geometry regimes. Through extensive curvature-stratified experiments, we provide a rigorous evaluation of whether GFMs can generalize uniformly across graphs with distinct intrinsic geometries.

A.6 Discussion: Why These Models?

In summary, our model selection reflects a curated spectrum of geometric assumptions rather than a mere collection of popular baselines. MLP removes graph geometry entirely; GCN, GAT, GraphSAGE, and PCNet operate in Euclidean space with different aggregation or filtering mechanisms; HGNN, HGCN, HAT, and HyboNet instantiate negative-curvature inductive bias; QGCN and CUSP model pseudo- or mixed-curvature structures; GraphMoRE adapts curvature at a finer granularity; and graph foundation models test whether pretrained graph knowledge transfers across curvature regimes. This taxonomy adheres to the core evaluation principle of CurvBench: model performance must be interpreted through the lens of graph geometry. Consequently, rather than seeking a universally superior model, our analysis investigates the specific curvature and task regimes where each inductive bias excels.

B Detailed Description of Datasets

To comprehensively evaluate various methods, we conduct experiments on a wide variety of public benchmark graph datasets. As outlined below, these datasets encompass diverse semantic contexts and topological structures:

- **Citation Networks (*Cora*, *Citeseer*, *PubMed*)**: Standard academic citation graphs where nodes represent scientific papers and edges denote undirected citations. Node features are bag-of-words representations of the documents, and labels correspond to academic subfields.
- **Webpage and Wikipedia Networks (*Cornell*, *Actor*)**: *Cornell* consists of web pages from Cornell University with hyperlinks as edges. *Actor* is an actor co-occurrence network constructed from Wikipedia pages, which typically exhibits distinct heterophilic properties.
- **Domain-Specific and Social Networks**:
  - *Airport*: A transportation network where nodes are airports and edges indicate flights between them.
  - *Disease*: An epidemiological network designed to simulate disease propagation dynamics.
  - *Telecom*: A telecommunications network modeling interactions and connectivity between customer entities.

– *CS_Phds*: An academic social network describing computer science PhD students and their advising/collaboration relationships.

Data Preprocessing Details. Given the varied sources of the raw data, all datasets are systematically structured into a unified standard tensor format (e.g., PyTorch Geometric `Data` objects) containing node feature matrices $\mathbf{X}$, adjacency edge indices $\mathbf{A}$, and label vectors $\mathbf{y}$. Due to scale contradictions in raw feature measurements, we enforce a global row-wise feature normalization technique seamlessly across all samples to prevent numerical overflows and computational instability during topological metrics calculations.

Label Processing and Splitting Protocol. Translating specialized datasets such as ***Airport*** and ***CS_Phds*** into robust categorical evaluation frames inherently requires addressing domain-specific challenges through several crucial data transformation procedures:

- ***Airport* Data Treatment:** This raw transportation network intrinsically comprises hundreds of unverified or metadata-missing airports mapped strictly to a dummy label class (i.e., -1). During the split sampling formulation, these non-labeled entities are unequivocally systematically purged from the label masking pools. They function exclusively as structural connective bridges propagating topological message passing but explicitly evade calculating classification loss or generalizability measurements.
- ***CS_Phds* Quartile Discretization:** In contrast to standard graphs, *CS_Phds* fundamentally originates as a continuous regression task predicting researchers' academic scalability metrics. To seamlessly interoperate within our 4-class node classification benchmark, we executed targeted uniform quartile discretization on its labels. By stratifying the inherent continuous values, we coerced the distribution into 4 distinct, relatively perfectly balanced numeric intervals (each absorbing roughly 250 samples uniformly).

Upon rectifying the labels array layout correspondingly, a tailored dataset stratification was implemented on *Airport* and *CS_Phds*. Since neither offers predefined independent splits, their training subsets are constituted adhering precisely to an internally seeded *stratified dynamic K-shot sampling schema*. Remaining categorized nodes from each class are subsequently divided orthogonally between isolated `validation` and `test` boolean masks.

Construction of the Telecommunication Network (*Telecom*). The *Telecom* dataset represents a real-world physical communication infrastructure, which is programmatically constructed from multi-relational tabular logs. In this network, nodes correspond to distinct telecommunication network elements and hardware equipment, such as base stations (`NODEB`), microwave transmission devices (`MICROWAVE`), and core routers (`ROUTER`). The edges are established based on the physical cabling or logical communication links that connect these network elements in the relational databases. To formulate the node features, equipment-specific configurations and heterogeneous categorical attributes (spanning up to 240 property columns in the raw tabular data) are sequentially extracted, numerically encoded, and flattened into 240-dimensional continuous feature vectors for each node. Ultimately, the tabular records and their interrelated equipment dependencies are projected into a unified graph structure, yielding a highly sparse infrastructure topology comprising 41,143 nodes and 41,424 edges.

Automated Table-to-Graph Construction Pipeline. To structurally bridge the gap between multi-relational databases and graph-based computational operations, we orchestrate an automated Table-to-Graph conversion framework that flawlessly preserves both topological schemas and tabular semantics. Guided by explicitly predefined Entity-Relationship (ER) mappings extracted from the original database architectures (e.g., matching foreign keys such as projecting chemical bonds to atoms, or biopsy records to specific patients), we project each isolated data table as a distinct node type within a PyTorch Geometric `HeteroData` object. For node feature initialization, we abstain from elementary hashing of table contents; instead, we deploy the CARTE framework, leveraging a pre-trained semantic language model (*FastText*), to sequentially embed the heterogeneous row-level properties—spanning numerical, categorical, and textual columns—into continuous, dense feature representations. Subsequently, the heterogeneous graph topology is systematically instantiated by performing intersection alignments across primary and foreign keys, translating discrete tabular cross-references into explicitly weighted adjacency edge indexes seamlessly.

Data splits for GFMs. As shown in Table 5, we adopt a geometry-balanced three-fold splitting protocol over the nine natural graph datasets for GFM evaluation. The datasets are first grouped into three curvature regimes: near-zero, positive, and negative. In each fold, we select two datasets from each geometry regime as source datasets for pre-training, and hold out the remaining dataset in the same regime for evaluation.

As a result, every fold contains six pre-training datasets and three evaluation datasets, with exactly one held-out dataset from each curvature regime. This design ensures that the pre-training stage always observes geometrically diverse source graphs, while the evaluation stage tests whether the learned transferable representations generalize to unseen datasets under each curvature regime. Across the three folds, every dataset is used once as an evaluation dataset and twice as a pre-training dataset, yielding a balanced protocol for assessing geometry-conditioned transfer behavior.

Table 5: Data splits for Graph Foundation Models.

Dataset	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Telecom	Disease	CS_Phds
Fold A	Evaluate	Pretrain	Pretrain	Evaluate	Pretrain	Pretrain	Evaluate	Pretrain	Pretrain
Fold B	Pretrain	Evaluate	Pretrain	Pretrain	Evaluate	Pretrain	Pretrain	Evaluate	Pretrain
Fold C	Pretrain	Pretrain	Evaluate	Pretrain	Pretrain	Evaluate	Pretrain	Pretrain	Evaluate

C Implementation Details

To ensure a faithful and fair evaluation, we maintain consistent experimental protocols across all considered baselines. Unless otherwise specified, all results are reported as averages over five independent runs with distinct random seeds. Experiments are primarily conducted on a single NVIDIA A100 GPU. For conventional graph models, a 24GB GPU is sufficient for reproduction. However, due to the substantial memory overhead of the pre-training stage, GFM baselines are evaluated on 80GB configurations. While hardware transitions may introduce minor runtime variations, they do not impact the reported performance metrics or the resulting scientific conclusions. All methods are evaluated under identical hyperparameter settings whenever applicable to ensure parity. Comprehensive implementation details, including specific training protocols and step-by-step reproduction instructions, are provided in our released code repository.

D Proofs for Section 4

D.1 Proof of Theorem 4.1

Proof. For a generic metric space $(\mathcal{X}, d_{\mathcal{X}})$ and a valid quadruple $q = (a, b, c; m)$, write

$$x = d_{\mathcal{X}}(a, m), \qquad y = d_{\mathcal{X}}(b, c), \qquad z = d_{\mathcal{X}}(a, b), \qquad w = d_{\mathcal{X}}(a, c).$$

Define

$$\Phi(x, y, z, w) = \frac{x^2 + \frac{1}{4}y^2 - \frac{1}{2}(z^2 + w^2)}{2x} = \frac{x}{2} + \frac{y^2}{8x} - \frac{z^2}{4x} - \frac{w^2}{4x}.$$

Then $\xi_{\mathcal{X}}(q) = \Phi(x, y, z, w)$. Moreover, Φ is homogeneous of degree one:

$$\Phi(\lambda x, \lambda y, \lambda z, \lambda w) = \lambda \Phi(x, y, z, w).$$

Let (x_G, y_G, z_G, w_G) be the four graph distances and (x_M, y_M, z_M, w_M) be the corresponding distances after embedding into $\mathcal{M}$. By the definition of $\delta = \mathrm{dis}_{\infty}^{\lambda}(h; G, \mathcal{M})$,

$$|x_M - \lambda x_G| \le \delta, \quad |y_M - \lambda y_G| \le \delta, \quad |z_M - \lambda z_G| \le \delta, \quad |w_M - \lambda w_G| \le \delta.$$

Because q is valid, $x_G = d_G(a, m) \ge 1$. Since $\delta < \lambda/2$,

$$x_M \ge \lambda x_G - \delta \ge \lambda - \delta > \lambda/2.$$

Moreover, every finite graph distance is at most $D = \mathrm{diam}_f(G)$, so

$$x_M, y_M, z_M, w_M \le \lambda D + \delta \le \lambda(D + 1/2).$$

It remains to bound the Lipschitz constant of Φ on the region

$$x \geq \lambda/2, \qquad 0 \leq x, y, z, w \leq \lambda(D + 1/2).$$

The partial derivatives are

$$\frac{\partial \Phi}{\partial x} = \frac{1}{2} - \frac{y^2}{8x^2} + \frac{z^2 + w^2}{4x^2}, \qquad \frac{\partial \Phi}{\partial y} = \frac{y}{4x},$$
$$\frac{\partial \Phi}{\partial z} = -\frac{z}{2x}, \qquad \frac{\partial \Phi}{\partial w} = -\frac{w}{2x}.$$

On the above region,

$$\left|\frac{\partial \Phi}{\partial x}\right| \leq \frac{1}{2} + \frac{5}{8}(2D + 1)^2,$$

and

$$\left|\frac{\partial \Phi}{\partial y}\right| + \left|\frac{\partial \Phi}{\partial z}\right| + \left|\frac{\partial \Phi}{\partial w}\right| \leq \frac{5}{4}(2D + 1).$$

Therefore, by the mean value theorem and the ℓ_∞ bound on the four distance errors,

$$|\Phi(x_M, y_M, z_M, w_M) - \Phi(\lambda x_G, \lambda y_G, \lambda z_G, \lambda w_G)| \leq C_D \delta,$$

where

$$C_D = \frac{1}{2} + \frac{5}{8}(2D + 1)^2 + \frac{5}{4}(2D + 1).$$

Using homogeneity,

$$\Phi(\lambda x_G, \lambda y_G, \lambda z_G, \lambda w_G) = \lambda \Phi(x_G, y_G, z_G, w_G) = \lambda \xi_G(q).$$

Hence,

$$|\xi_{\mathcal{M}}(h(q)) - \lambda \xi_G(q)| \leq C_D \delta.$$

Taking expectation over $q \sim \mathcal{Q}_G$ gives Eq. (9).

Finally, let

$$D_{\mathcal{H}}^{\lambda} = \mathrm{Dist}_{\mathcal{H}}^{\lambda}(G, \mathcal{M}), \qquad \eta_{\mathcal{H}}^{\lambda} = \eta_{\mathcal{H}}^{\lambda}(G, \mathcal{M}).$$

If $D_{\mathcal{H}}^{\lambda} < \lambda/2$, then for any sufficiently small $\rho > 0$, there exists $h_\rho \in \mathcal{H}$ such that

$$\mathrm{dis}_{\infty}^{\lambda}(h_\rho; G, \mathcal{M}) \leq D_{\mathcal{H}}^{\lambda} + \rho < \lambda/2.$$

Applying Eq. (9) to h_ρ gives

$$\eta_{\mathcal{H}}^{\lambda} \leq C_D \left(D_{\mathcal{H}}^{\lambda} + \rho\right).$$

Letting $\rho \to 0$ yields

$$D_{\mathcal{H}}^{\lambda} \geq \frac{\eta_{\mathcal{H}}^{\lambda}}{C_D}.$$

Thus proves the theorem.

D.2 Proof of Theorem 4.2

Proof. Fix a model pair (i, j) and write $p_r^z = p_{ij,r}^z$ for simplicity. If two graphs are drawn independently from the same regime r, then the probability that their pairwise states agree is

$$\sum_{z \in \{-1,0,+1\}} (p_r^z)^2.$$

Thus the within-regime disagreement probability for this pair is

$$D_{ij}^{\mathrm{within}} = 1 - \frac{1}{K} \sum_{r \in \mathcal{R}} \sum_{z \in \{-1,0,+1\}} (p_r^z)^2.$$

If the two graphs are drawn independently from two different regimes, then the cross-regime agreement probability is

$$\frac{1}{K(K-1)} \sum_{r \neq s} \sum_{z \in \{-1,0,+1\}} p_r^z p_s^z.$$

Therefore

$$D_{ij}^{\text{cross}} = 1 - \frac{1}{K(K-1)} \sum_{r \neq s} \sum_{z \in \{-1,0,+1\}} p_r^z p_s^z.$$

Subtracting the two quantities gives

$$D_{ij}^{\text{cross}} - D_{ij}^{\text{within}} = \sum_{z \in \{-1,0,+1\}} \left[\frac{1}{K} \sum_r (p_r^z)^2 - \frac{1}{K(K-1)} \sum_{r \neq s} p_r^z p_s^z \right].$$

For each fixed state z, define

$$\bar{p}^z = \frac{1}{K} \sum_r p_r^z, \qquad \overline{(p^z)^2} = \frac{1}{K} \sum_r (p_r^z)^2.$$

Since

$$\sum_{r \neq s} p_r^z p_s^z = \left(\sum_r p_r^z \right)^2 - \sum_r (p_r^z)^2,$$

we have

$$\frac{1}{K} \sum_r (p_r^z)^2 - \frac{1}{K(K-1)} \sum_{r \neq s} p_r^z p_s^z = \frac{K}{K-1} \left(\overline{(p^z)^2} - (\bar{p}^z)^2 \right).$$

The term in parentheses is $\text{Var}_{r \sim \text{Unif}(\mathcal{R})}(p_{ij,r}^z)$. Hence

$$D_{ij}^{\text{cross}} - D_{ij}^{\text{within}} = \frac{K}{K-1} \sum_{z \in \{-1,0,+1\}} \text{Var}_r \left(p_{ij,r}^z \right).$$

Finally, $d_\epsilon(G, G')$ is the average of pairwise disagreement indicators over all $\binom{N}{2}$ model pairs. Averaging the pairwise identity over $i < j$ proves Eq. (15). The non-negativity follows from non-negativity of variance. The gap is zero if and only if all these variances are zero, i.e., if and only if every pairwise comparison-state distribution is invariant across regimes.

E Details of Metrics

This section details the shortest-path four-point residual and the skewness of its node-level distribution. The residual is a midpoint-inspired metric descriptor; because arbitrary neighbor pairs need not have an exact midpoint at the center, its sign is not a direct estimator of manifold sectional curvature or a substitute for edge Ricci curvatures. Skewness describes asymmetry in the resulting node scores.

E.1 Shortest-path four-point residual

Let $G = (V, E)$ be an undirected graph with adjacency matrix A and pairwise distance matrix D, where $D_{uv} = d_G(u, v)$ denotes the graph distance between nodes u and v. In our implementation, D is typically instantiated as the all-pairs shortest-path distance matrix. For disconnected graphs, invalid or infinite distances are excluded from the corresponding averaging operations.

For a center node $m \in V$, we consider unordered neighbor pairs

$$\mathcal{P}_m = \big\{ \{b, c\} : b, c \in \mathcal{N}(m),\ b < c \big\},$$

where $\mathcal{N}(m) = \{v \in V : (m, v) \in E\}$ is the one-hop neighborhood of m. Each pair $\{b, c\}$ defines a local metric section around m. To probe this section, we further introduce an anchor node $a \in V \setminus \{m\}$ satisfying $0 < d_G(a, m) < \infty$. For every valid quadruple $(a, b, c; m)$, we define the midpoint curvature residual as

$$\Delta_G(a, b, c; m) = d_G(a, m)^2 + \frac{1}{4} d_G(b, c)^2 - \frac{1}{2} \Big(d_G(a, b)^2 + d_G(a, c)^2 \Big).$$

The normalized four-point residual is then given by

$$\xi_G(a, b, c; m) = \frac{d_G(a, m)^2 + \frac{1}{4} d_G(b, c)^2 - \frac{1}{2} \left(d_G(a, b)^2 + d_G(a, c)^2 \right)}{2 d_G(a, m)}. \tag{16}$$

The Euclidean midpoint identity gives zero when m is the actual midpoint of b and c. A graph neighbor pair need not satisfy that condition, even if all edges have unit length. For instance, in the ordinary square grid with graph (Manhattan) distance, $m = (0, 0)$, $b = (1, 0)$, $c = (-1, 0)$, and $a = (0, 1)$ give $d(a, m) = 1$, $d(b, c) = 2$, and $d(a, b) = d(a, c) = 2$. Equation (16) then yields $\xi_G = -1$, although a square grid is often treated as a flat example. This is an analytic counterexample to interpreting every negative residual as intrinsically hyperbolic. Our diagnostic uses the distribution of this fixed score consistently across the reported graphs; alternative metrics such as balanced Forman curvature and effective resistance probe different aspects of topology (Topping et al., 2022; Black et al., 2023; **?**). No equivalence or measured correlation between these notions is claimed.

The raw curvature estimate of node m is obtained by averaging the residual over all valid anchor nodes and all unordered neighbor pairs:

$$\widehat{\kappa}_G(m) = \frac{1}{|\mathcal{P}_m|} \sum_{\{b,c\} \in \mathcal{P}_m} \frac{1}{|\mathcal{A}_m|} \sum_{a \in \mathcal{A}_m} \xi_G(a, b, c; m), \tag{17}$$

where

$$\mathcal{A}_m = \{a \in V \setminus \{m\} : 0 < d_G(a, m) < \infty\}.$$

When $|\mathcal{P}_m| = 0$, i.e., when the degree of m is smaller than two, we set $\widehat{\kappa}_G(m) = 0$. This convention avoids introducing artificial curvature signals for nodes whose local metric section cannot be formed.

For cross-dataset comparability, we further use a relative curvature normalization:

$$\kappa_G(m) = \frac{\widehat{\kappa}_G(m)}{\mathrm{diam}_f(G)}, \qquad \mathrm{diam}_f(G) = \max_{u,v:\, d_G(u,v) < \infty} d_G(u, v). \tag{18}$$

This normalization reduces direct dependence on graph diameter; it does not by itself make graphs of different size, density, or sampling structure geometrically equivalent.

Implementation details. Our curvature computation is implemented as a GPU-accelerated tensor routine. Given an adjacency matrix A and a corresponding distance matrix D, the algorithm iterates over center nodes m to extract neighbor sets, forms unordered pairs $\{b, c\}$, and evaluates Eq. (16) across all valid anchors a. To ensure scalability and reliability, the implementation offers two operational modes:

`Fast` Mode: Utilizes lightweight, chunked computation designed for medium-scale graphs, prioritizing throughput.

`Strict` Mode: Employs conservative memory access and accumulates statistics in higher precision, making it the preferred choice for large-scale experiments and final reporting.

To manage memory overhead, both modes utilize pair-wise chunking to avoid the simultaneous materialization of all anchor-pair combinations. Furthermore, the pipeline automatically filters invalid denominators and infinite distances. Once node-level curvature values are computed, we apply relative normalization based on the maximum finite entry in the distance matrix to ensure stability across different graph scales.

E.2 Curvature Skewness

The mean curvature of a graph provides a first-order summary of its global geometric tendency:

$$\overline{\kappa}(G) = \frac{1}{|V|} \sum_{m \in V} \kappa_G(m).$$

However, mean curvature alone is insufficient for characterizing heterogeneous relational data. A graph may have near-zero average curvature while still containing a small subset of nodes with strongly positive or strongly negative curvature. Such tail behavior is especially important in table-derived graphs, where foreign-key joins, hub entities, and relational schema construction can induce highly asymmetric local geometry. Therefore, CURVBENCH also computes the skewness of the node-level curvature distribution.

Let

$$\sigma_\kappa(G) = \left(\frac{1}{|V|} \sum_{m \in V} \left(\kappa_G(m) - \overline{\kappa}(G)\right)^2 \right)^{1/2}$$

be the standard deviation of node-level curvature values. The curvature skewness is defined as the third standardized central moment:

$$\gamma_\kappa(G) = \begin{cases} \dfrac{1}{|V|} \displaystyle\sum_{m \in V} \left(\dfrac{\kappa_G(m) - \overline{\kappa}(G)}{\sigma_\kappa(G)} \right)^3, & \sigma_\kappa(G) > 0, \\ 0, & \sigma_\kappa(G) = 0. \end{cases}$$

Positive skewness indicates a right-tailed curvature distribution, where a minority of nodes exhibit substantially higher positive curvature than the graph average. Negative skewness indicates a left-tailed distribution, where a minority of nodes exhibit substantially lower curvature. Values close to zero suggest a comparatively balanced curvature profile.

Implementation details. In the implementation, node-level curvature tensors are first loaded from saved `.pt` files. Non-finite entries are removed before computing statistics. The skewness is then computed directly as

$$\text{mean}\left[\left(\frac{\kappa - \mu}{\sigma} \right)^3 \right],$$

where μ and σ are the empirical mean and standard deviation of finite node curvature values.

E.3 Why Both Metrics Are Needed

The synergy between mean curvature $\overline{\kappa}(G)$ and curvature skewness $\gamma_\kappa(G)$ is fundamental to CurvBench. While mean curvature captures a dataset's average signed geometry, it often masks local heterogeneity. Skewness complements this by revealing whether the curvature distribution is dominated by asymmetric tails—a distinction critical to relational learning, where performance is often driven by rare but influential geometric regions rather than just the global average.

Geometric Nuance and Inductive Bias. This dual-metric approach prevents the misclassification of complex topologies:

Positive Skewness ($\gamma_\kappa(G) \gg 0$): A graph with $\overline{\kappa}(G) \approx 0$ may appear Euclidean, yet a high positive skew indicates a strong positive-curvature tail. Such structures favor models designed for compact, clustered, or schema-induced patterns.

Negative Skewness ($\gamma_\kappa(G) < 0$): Conversely, negative skewness points to tree-like or hierarchically expanding regions, which align more effectively with hyperbolic or mixed-curvature inductive biases.

Relying solely on $\overline{\kappa}(G)$ would collapse these distinct geometries into a single regime, inadvertently reproducing the same failure modes found in conventional aggregate benchmarks.

The Role of Metrics in CurvBench. Within the CurvBench framework, these metrics fulfill two primary functions:

Interpretability: They provide a pre-training geometric summary of each dataset.

Stratification: They facilitate regime-stratified evaluation, conditioning model comparisons on the data's geometric profile.

By moving away from arbitrary dataset mixtures, CurvBench shifts the focus from identifying a "globally best" model to determining which inductive biases are most effective under specific geometric conditions.

E.4 Ranking consistency metrics

To quantify whether datasets within the same geometry regime induce similar model preferences, we compare the model rankings produced by different datasets using Spearman correlation, Kendall rank correlation, and top-k Jaccard overlap.

Let $\mathcal{M} = \{M_1, \ldots, M_N\}$ denote the set of evaluated models. For a dataset G, let $r_G(M_i)$ be the rank position of model M_i, where a smaller value indicates better performance. In our top-3

truncated ranking protocol, the top three models retain their exact ranks, while all remaining models are assigned the same lower-tier rank. This focuses the comparison on the most competitive models while treating lower-ranked methods as indistinguishable.

Spearman correlation. Spearman correlation measures the global monotonic agreement between two rankings. Given two datasets G and G', we compute Spearman correlation as the Pearson correlation between their rank vectors:

$$\rho(G, G') = \frac{\sum_{i=1}^{N} \left(r_G(M_i) - \bar{r}_G\right)\left(r_{G'}(M_i) - \bar{r}_{G'}\right)}{\sqrt{\sum_{i=1}^{N} \left(r_G(M_i) - \bar{r}_G\right)^2}\sqrt{\sum_{i=1}^{N} \left(r_{G'}(M_i) - \bar{r}_{G'}\right)^2}}, \tag{19}$$

where

$$\bar{r}_G = \frac{1}{N}\sum_{i=1}^{N} r_G(M_i), \qquad \bar{r}_{G'} = \frac{1}{N}\sum_{i=1}^{N} r_{G'}(M_i).$$

A larger $\rho(G, G')$ indicates that models ranked highly on G also tend to be ranked highly on G'. Thus, Spearman correlation captures global consistency over the whole ranked model list.

Kendall rank correlation. Kendall correlation measures pairwise ordering agreement between two rankings. For each model pair (M_i, M_j), we compare whether the relative order of the pair is preserved across two datasets. Define

$$s_G(i, j) = \operatorname{sign}\left(r_G(M_j) - r_G(M_i)\right),$$

where $s_G(i, j) = +1$ means M_i ranks above M_j, $s_G(i, j) = -1$ means M_j ranks above M_i, and $s_G(i, j) = 0$ means the two models are tied. The Kendall correlation can then be written as

$$\tau(G, G') = \frac{\sum_{i<j} s_G(i, j) s_{G'}(i, j)}{\sqrt{\sum_{i<j} s_G(i, j)^2}\sqrt{\sum_{i<j} s_{G'}(i, j)^2}}. \tag{20}$$

This formulation naturally handles ties induced by the top-3 truncation. A high Kendall correlation means that pairwise model preferences are stable: if M_i outperforms M_j on one dataset, the same ordering is likely to hold on the other dataset.

Top-k Jaccard overlap. While Spearman and Kendall correlations compare ranked lists, Jaccard overlap compares the identity of the top-performing models. Let

$$T_k(G) = \{M_i : M_i \text{ is among the top-}k \text{ models on } G\}.$$

The top-k Jaccard overlap between two datasets is defined as

$$J_k(G, G') = \frac{|T_k(G) \cap T_k(G')|}{|T_k(G) \cup T_k(G')|}. \tag{21}$$

In our experiments, we set $k = 3$. Thus, $J_3(G, G')$ measures whether two datasets share the same leading models, regardless of the exact ordering among lower-ranked methods.

Together, these metrics evaluate ranking consistency at different levels. Spearman correlation measures global monotonic agreement, Kendall correlation measures pairwise preference stability, and top-3 Jaccard overlap measures agreement among the strongest models. If within-regime similarity is consistently higher than cross-regime similarity under all three metrics, this indicates that datasets in the same curvature regime induce coherent model preferences.

F Related Work

Relational Learning. Relational learning aims to model data instances connected through explicit or implicit dependencies, covering a broad range of domains such as citation networks, molecular graphs, social systems, knowledge graphs, and table-derived relational data Battaglia et al. (2018); Fey et al. (2023); Sun et al. (2026b). Although these datasets differ substantially in scale, sparsity, feature modality, and task formulation, they share a common structural abstraction: relations induce paths, neighborhoods, and higher-order connectivity patterns that define an underlying topology Eremeev et al. (2025); Du et al. (2026). This has motivated the development of unified relational learning

benchmarks, where diverse datasets are collected to evaluate model generalization across domains. However, most existing benchmarks summarize model performance through flat leaderboards that average results across heterogeneous datasets Lv et al. (2021). Such an evaluation protocol implicitly assumes that relational data forms a structurally uniform category, while in practice different datasets may exhibit fundamentally different geometric properties. As a result, aggregate rankings can obscure regime-specific strengths and weaknesses, making it difficult to determine whether a model is genuinely robust or merely well aligned with the dominant structure of a benchmark.

Benchmarks for relational learning. Standardized benchmarks—most notably OGB, Benchmarking GNNs, and GraphGym—have served as the bedrock of relational learning by formalizing reproducibility through common datasets and evaluation protocols (Hu et al., 2020; Dwivedi et al., 2023; You et al., 2020). Recent expansions have even extended these frameworks to relational databases and table-derived graphs, mapping rows and foreign-key relations into structured graph formats (Fey et al., 2023; Kim et al., 2024). However, a fundamental limitation persists: most benchmarks rely on flat leaderboards that aggregate performance across heterogeneous datasets. While useful for coarse comparisons, these global averages implicitly assume a degree of structural uniformity that rarely exists in practice. Datasets vary not only in scale and homophily but also in intrinsic metric geometry. Consequently, aggregate rankings often obscure regime-dependent behaviors, making model superiority appear more universal than the underlying data geometry justifies. CURVBENCH addresses this by introducing curvature as an explicit evaluation axis, shifting the paradigm from global rankings to regime-conditioned analysis. Recent work on relational-learning benchmarks has further highlighted the lack of standardized comparison and aggregation protocols across heterogeneous tasks and datasets (Hayler et al., 2026).

Non-Euclidean Graph Learning and Geometry-Aware Evaluation. Research has increasingly demonstrated that graph representation learning is heavily influenced by the geometry of the embedding space. While Euclidean-based models like GCN, GAT, and GraphSAGE remain robust baselines (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017), Euclidean space is often fundamentally misaligned with hierarchical, tree-like, or mixed-structure graphs. This has motivated the rise of hyperbolic neural networks and embeddings designed for negatively curved spaces (Nickel & Kiela, 2017; Ganea et al., 2018), resulting in specialized architectures such as HGCN and hyperbolic attention mechanisms (Chami et al., 2019; Zhang et al., 2021a).

To better match non-Euclidean structures Asif et al. (2021), a growing body of work has introduced multiple experts into graph learning. These methods explicitly encode geometric inductive biases, aiming to better capture hierarchical, clustered, or heterogeneous graph structures Ye et al. (2025); Wang et al. (2025a). More recent advances in pseudo-Riemannian, mixed-curvature, and adaptive Riemannian models further relax the assumption of a static geometry (Xiong et al., 2022; Guo et al., 2025a). Despite being predicated on the idea that representation geometry should match data geometry, these models are still typically assessed using the same flat averages as their Euclidean counterparts. CURVBENCH closes this gap by stratifying datasets via mean curvature and curvature skewness. This enables a rigorous assessment of when Euclidean, hyperbolic, adaptive, or foundation models are truly aligned with the underlying relational geometry.

G ADDITIONAL EXPERIMENTS AND OBSERVATIONS

This section provides supplementary analyses that build upon our primary findings. While the main text focuses on curvature-conditioned rankings and the reorganization of model families, we further examine four critical diagnostic dimensions: (i) Leaderboard Deviation: We quantify the degree to which each curvature-stratified leaderboard diverges from the conventional flat leaderboard. (ii) Task Consistency: We investigate whether curvature-conditioned preferences remain stable across diverse relational tasks. (iii) GFM Supervision and Feasibility: We analyze the interaction between additional supervision signals and the practical feasibility of deploying Graph Foundation Models. (iv) Specialist–Robustness Trade-offs: We evaluate whether table-derived graphs necessitate a trade-off between regime-specific expertise and general model robustness.

Rank-shift analysis across tasks and metrics. Using Tables **??**, 6, **??**, and 7, we compute the RSI for each curvature regime under four evaluation views: NC performance, NC Macro-F1, LP AUC,

Table 6: Macro-F1 results on Node Classification (NC) task. Datasets and baselines are divided into different regimes. We highlight the top-3 results with **red bolded**, red and **bold**.

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCN	79.36±1.13	65.64±0.66	77.90±0.29	66.10±1.16	24.88±3.07	22.74±1.02	59.19±2.59	57.19±0.46	26.47±0.84
GAT	80.02±0.45	64.43±1.03	76.69±0.25	**82.05±1.18**	27.71±2.35	22.63±1.20	**84.41±2.86**	52.51±0.15	10.58±0.00
GraphSAGE	**87.60±0.28**	71.59±0.53	88.09±0.06	29.24±0.56	**51.72±5.97**	**31.82±0.53**	91.50±3.04	**71.84±0.38**	10.47±1.95
MLP	54.86±1.06	54.18±0.87	71.41±0.71	85.90±0.81	**51.25±1.71**	**35.46±1.90**	44.41±0.00	58.45±0.03	10.58±0.00
PCNet	87.41±0.47	**73.15±0.15**	**89.30±0.14**	30.43±0.50	48.42±5.91	31.62±1.00	49.78±1.23	57.97±0.03	12.62±1.08
HGNN	57.51±0.38	58.09±0.42	73.02±0.66	24.56±3.17	51.54±5.24	33.72±0.60	39.16±2.92	58.18±0.04	18.84±2.51
HAT	**80.33±0.31**	**67.79±0.32**	**78.27±0.42**	58.55±6.60	11.34±0.13	26.36±0.52	43.67±0.00	58.24±0.01	10.57±0.00
HGCNN	77.37±0.73	65.03±0.43	76.33±0.47	43.60±5.96	45.55±2.35	21.07±1.34	76.88±2.35	**61.96±8.29**	42.57±2.98
HyboNet	73.83±0.39	65.18±1.53	73.46±0.47	47.06±3.93	18.48±5.23	17.81±2.55	76.47±5.23	38.45±11.22	10.54±0.06
CUSP	76.12±1.17	64.22±1.00	65.99±2.16	47.86±2.71	13.53±0.00	13.33±3.16	43.74±19.17	36.18±6.88	12.77±3.52
QGCN	78.53±0.48	63.87±0.19	75.66±0.98	47.17±0.79	26.81±1.54	21.61±0.76	83.01±1.52	66.46±0.27	**43.86±2.53**
GraphMoRE	80.21±0.23	64.59±1.03	76.01±1.02	**90.47±1.19**	19.62±2.31	22.76±0.66	**93.80±1.34**	64.49±0.74	**36.12±2.94**

Table 7: AP results on Link Prediction (LP) task. Datasets and baselines are divided into different regimes. We highlight the top-3 results with **red bolded**, red and **bold**.

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCN	**92.18±1.18**	**93.16±0.56**	92.28±0.26	94.04±0.66	73.72±10.41	**82.59±0.63**	49.58±2.75	69.37±0.77	42.14±0.48
GAT	92.46±0.28	93.58±0.52	91.02±0.22	93.20±0.65	**73.80±6.49**	81.88±0.98	50.13±3.01	**69.79±0.86**	47.56±0.28
GraphSAGE	68.19±1.10	68.52±1.47	84.31±0.64	71.33±0.84	56.01±6.86	56.24±1.29	49.98±0.05	59.68±0.52	43.34±0.17
MLP	82.25±1.46	89.59±1.57	85.29±0.50	90.71±0.67	67.46±6.49	70.85±0.93	51.50±0.35	67.60±0.91	50.00±0.00
PCNet	73.20±0.52	68.87±2.03	94.05±0.13	71.98±0.73	56.44±5.16	63.74±1.04	58.12±8.37	67.22±0.42	**83.49±0.76**
HGNN	68.96±0.95	84.47±0.48	90.83±0.24	93.48±0.24	64.94±4.90	71.46±0.55	51.90±1.80	**68.81±0.16**	50.84±2.44
HGCNN	83.54±1.81	89.06±0.62	**94.01±0.07**	93.68±0.21	**71.16±3.02**	82.19±0.31	**66.54±6.12**	59.10±1.28	55.19±1.72
Fully-HNN	90.33±1.65	79.27±1.46	92.20±0.41	**96.16±0.70**	68.27±4.79	85.14±1.05	46.55±1.50	55.18±0.72	59.38±4.89
CUSP	88.07±1.55	90.12±1.86	60.04±0.90	73.24±0.97	61.89±9.41	72.81±1.45	36.87±1.85	64.71±0.90	**56.29±2.18**
GraphMoRE	**96.87±0.15**	**98.48±0.28**	**98.61±0.18**	**94.73±0.42**	64.71±3.16	**87.55±0.57**	65.21±2.31	68.30±0.54	46.86±3.16
QGCN	88.35±0.26	88.85±0.63	93.40±0.23	95.69±0.08	62.61±1.66	79.70±0.67	**64.29±1.20**	64.76±0.69	54.98±1.71

and LP AP. This multi-metric analysis avoids drawing conclusions from a single task-metric pair and allows us to examine whether the distortion induced by flat averaging is stable across evaluation criteria.

Table 8 shows that rank distortion is not uniform across tasks. For node classification, both performance and Macro-F1 lead to the same conclusion: flat averaging is most misleading in the near-zero and positive regimes. The average RSI is 2.92 in the near-zero regime and 2.50 in the positive regime, but only 1.25 in the negative regime. This indicates that, for NC, the global leaderboard hides substantial regime-specific reorganization among citation-like and positively curved graphs. For example, Tables **??** and 6 consistently show that PCNet and GraphSAGE dominate the near-zero regime, while MLP becomes highly competitive in the positive regime, especially under Macro-F1. Such behavior would be obscured if all datasets were collapsed into a single flat average.

The LP task exhibits the opposite pattern. Across both AUC and AP, the largest distortion appears in the negative regime, with an average RSI of 2.36, compared with 0.82 in the near-zero regime. This suggests that LP rankings are relatively stable on citation-like graphs, but become much more geometry-sensitive on negatively curved graphs. Tables **??** and 7 show that PCNet and QGCN become particularly strong in the negative regime, whereas GraphMoRE dominates the near-zero regime under both LP metrics. Therefore, the same flat leaderboard can fail in different ways depending on the downstream task: NC aggregation mainly hides near-zero and positive-regime shifts, while LP aggregation mainly hides negative-regime specialization.

Table 8: Rank-shift index (RSI) between flat and regime-conditioned rankings across tasks and metrics. Larger values indicate stronger deviation from the flat leaderboard.

Evaluation view	Near-zero	Positive	Negative
NC Performance	3.00	2.50	1.17
NC Macro-F1	2.83	2.50	1.33
LP AUC	0.91	1.45	2.18
LP AP	0.73	1.82	2.55
Task-wise NC average	2.92	2.50	1.25
Task-wise LP average	0.82	1.64	2.36

Observation G.1: Flat leaderboards hide task-dependent rank distortions. The multi-metric RSI results in Table 8 strengthen the central claim of CURVBENCH. Flat leaderboards do not

merely introduce random noise; they induce systematic and task-dependent distortions. For NC, the main distortion comes from feature- and classification-sensitive regimes, whereas for LP, the main distortion comes from negatively curved regimes where pairwise proximity and structural closure become more important. Thus, regime-conditioned evaluation is not only a more fine-grained reporting format, but changes the substantive conclusion about which inductive bias is reliable under which structural condition.

Within-task cross-metric consistency. We next examine whether different metrics within the same task induce similar model preferences. For NC, we compare the regime-wise rankings produced by Tables **??** and 6. For LP, we compare the regime-wise rankings produced by Tables **??** and 7. The results are shown in Table 9.

Table 9: Within-task consistency between two metrics under each curvature regime. NC compares Accuracy and Macro-F1; LP compares AUC and AP.

Task	Regime	Spearman	Kendall	Top-3 Jaccard
NC	Near-zero	0.937	0.848	1.000
NC	Positive	0.769	0.606	0.500
NC	Negative	0.860	0.758	0.500
LP	Near-zero	0.991	0.964	1.000
LP	Positive	0.836	0.673	0.500
LP	Negative	0.864	0.709	0.500

Table 9 shows that metric choice does not destroy the curvature-conditioned structure of the results. Within each task, the two metrics produce highly consistent rankings in the near-zero regime: NC achieves a Spearman correlation of 0.937 and LP achieves 0.991, with identical top-3 model sets in both cases. This indicates that citation-like graphs induce stable model preferences regardless of the precise metric used. However, consistency becomes weaker in the positive and negative regimes, where the top-3 Jaccard overlap drops to 0.5. This suggests that non-flat regimes are more sensitive not only to model geometry, but also to what aspect of performance is measured. For example, in LP, AUC and AP agree that the near-zero regime strongly favors GraphMoRE, GCN, and GAT, but differ more noticeably in the positive and negative regimes.

Cross-task order consistency. Finally, we analyze whether NC and LP induce similar model preferences under the same curvature regime. Instead of comparing only one NC metric against one LP metric, we average the ranking-consistency scores over all four NC–LP metric pairs, where results are summarized in Table 10. For rank-based statistics, we use the common model set shared by the corresponding NC and LP tables.

Table 10: Cross-task consistency between NC and LP rankings, averaged over all NC–LP metric pairs. Higher values indicate stronger agreement between task-induced model orders.

Regime	Spearman	Kendall	Top-3 Jaccard
Near-zero	0.011	0.127	0.200
Positive	0.066	0.036	0.275
Negative	0.352	0.255	0.425

Table 10 shows that cross-task agreement is much weaker than within-task cross-metric agreement. In the near-zero regime, NC and LP have almost no global rank correlation, even though both tasks are evaluated on the same citation-like datasets. Tables **??** and 6 show that NC favors PCNet and GraphSAGE, whereas Tables **??** and 7 show that LP favors GraphMoRE together with strong Euclidean baselines such as GCN and GAT. This indicates that near-zero geometry alone does not determine a universal model order; the task objective determines which structural signal becomes useful.

The negative regime exhibits the strongest cross-task agreement, with Spearman correlation increasing to 0.352 and top-3 Jaccard overlap to 0.425. This is because both NC and LP repeatedly favor geometry-aware or structure-sensitive methods in negatively curved graphs, especially GraphMoRE, QGCN, HGCNN, and PCNet, although their exact ordering still differs by task and metric. The positive regime remains intermediate and metric-sensitive: MLP is highly competitive for NC, while LP favors methods such as GAT, GraphMoRE, and the hyperbolic baseline. Thus, curvature provides a meaningful structural context, but the final ranking is determined by the interaction among geometry, task objective, and evaluation metric.

Observation G.2: Model effectiveness is jointly shaped by geometry, task, and metric. The combined analysis of Tables 8, 9, and 10 refines the interpretation of CURVBENCH. Curvature is not a standalone oracle that assigns one universal ranking to each regime. Instead, it defines the structural

context in which task-specific and metric-specific preferences emerge. Within a fixed task, different metrics generally preserve the same broad curvature-conditioned trends, especially in the near-zero regime. Across tasks, however, rankings can diverge substantially because NC emphasizes feature separability, neighborhood aggregation, and label smoothness, whereas LP emphasizes pairwise proximity, structural closure, and long-range connectivity. Therefore, model effectiveness is jointly shaped by intrinsic geometry, downstream objective, and evaluation metric. This is precisely why CURVBENCH replaces a single flat leaderboard with regime-conditioned diagnostics: it reveals not only which model performs well, but also under which geometric and task-metric conditions its inductive bias becomes effective.

Table 11: Macro-F1 results of GFMs under 1-shot and 5-shot scenarios. OOM means Out-Of-Memory.

1-shot scenario

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCOPE	30.57±6.40	29.65±10.77	33.28±7.39	**18.67±7.17**	**24.03±6.35**	14.55±5.14	**43.21±2.38**	**41.49±6.56**	22.07±4.64
MDGPT	**42.75±8.24**	34.86±9.80	**51.89±11.34**	6.65±5.31	**28.14±7.47**	15.92±3.44	43.85±9.81	**33.42±8.98**	22.78±2.57
MDGFM	42.53±5.32	**37.66±6.14**	50.66±10.12	14.51±2.83	29.28±6.17	17.74±2.12	37.44±4.52	OOM	23.62±2.23
SAMGPT	25.01±8.42	**33.37±7.73**	39.46±9.91	13.35±6.17	**30.05±7.22**	**17.92±5.71**	**43.70±5.65**	34.74±8.46	**23.77±6.50**
GraphGluing	12.98±6.47	18.03±9.63	33.07±5.72	18.34±2.79	19.21±4.26	8.28±0.16	**47.39±4.25**	OOM	19.42±6.71
SA2GFM	**37.97±8.36**	26.45±6.44	**42.33±9.73**	**17.23±3.96**	18.50±3.40	**15.94±2.06**	42.67±6.90	OOM	**23.16±3.89**

5-shot scenario

Baselines	Cora	Citeseer	PubMed	Airport	Cornell	Actor	Disease	Telecom	CS_Phds
GCOPE	**60.44±2.06**	**50.77±5.38**	56.67±1.67	**15.84±2.65**	**64.78±6.99**	**21.63±2.38**	**55.91±7.32**	**55.04±5.59**	25.74±2.02
MDGPT	60.53±4.89	54.29±6.52	**58.41±7.05**	9.83±3.17	**41.18±6.58**	**19.67±4.09**	50.89±9.23	**38.23±7.64**	26.16±2.25
MDGFM	**64.31±4.11**	**55.28±4.37**	**64.99±5.11**	**16.10±2.13**	55.80±6.06	20.10±1.75	59.27±7.68	OOM	**25.95±2.46**
SAMGPT	35.88±6.95	50.26±5.89	55.08±9.11	15.23±4.77	**48.68±6.70**	17.91±6.14	**65.49±9.55**	43.09±7.45	**26.17±6.30**
GraphGluing	47.55±5.78	39.55±2.65	64.52±1.73	18.90±3.41	19.70±1.79	9.59±2.37	49.27±4.79	OOM	15.47±3.99
SA2GFM	49.24±6.34	35.41±4.19	50.63±9.67	**19.24±5.00**	17.07±4.52	13.52±2.57	48.92±7.84	OOM	**26.63±1.67**

GFM label elasticity and feasibility. We next analyze how GFMs respond to additional supervision under different geometric regimes. The raw 1-shot and 5-shot results are reported in Table 11. For model M and regime r, we define label elasticity as

$$\text{Elasticity}(M, r) = \frac{1}{|\mathcal{D}_r(M)|} \sum_{G \in \mathcal{D}_r(M)} \left(S_{M,G}^{\text{5-shot}} - S_{M,G}^{\text{1-shot}} \right), \tag{22}$$

where $\mathcal{D}_r(M)$ contains the datasets in regime r on which M is feasible. This quantity measures how much additional supervision can compensate for the inductive bias transferred from pretraining. High elasticity indicates that a model is mainly supervision-limited in that regime, whereas low or negative elasticity suggests that the bottleneck lies beyond label scarcity, such as geometric mismatch, optimization difficulty, or scalability constraints.

From Table 11, averaged over GFMs, the elasticity is 18.41 in the near-zero regime, 6.47 in the positive regime, and 6.97 in the negative regime. This indicates that near-zero citation-like graphs are highly label-elastic: once additional labels are provided, most GFMs improve substantially. In contrast, positive and negative regimes exhibit much weaker elasticity, suggesting that additional labels alone do not fully resolve the difficulty of these graphs. Their performance is instead shaped by a three-way interaction among geometry, task objective, and the transfer mechanism encoded by each GFM.

At the model level, Table 11 shows that GCOPE has the strongest average elasticity among complete-coverage methods, improving by 16.59 points on average while remaining feasible on all nine natural graphs. MDGFM has comparable elasticity (13.55) but fails on Telecom, reducing its complete-coverage reliability. GraphGluing is especially revealing: it gains 29.18 points in the near-zero regime but only 0.79 in the positive regime and slightly decreases in the negative regime. This suggests a geometry–supervision–scalability frontier. A GFM may benefit strongly from additional labels when its transferred representation is structurally compatible with the target regime, but the same mechanism may fail to produce gains when the regime requires different geometric reasoning or exceeds the model's feasible operating range.

Coverage-aware GFM comparison. Table 11 also shows that several GFMs encounter OOM on Telecom, making it necessary to distinguish accuracy from feasibility. To make the effect of OOM

explicit, we compare available-case averages with coverage-aware averages. The available-case average ignores OOM entries, whereas the coverage-aware average divides the total score by all nine datasets, thereby penalizing infeasible evaluations:

$$S_{\text{cov}}(M) = \frac{1}{9} \sum_{G} \mathbb{1}[M \text{ is feasible on } G] \cdot S_{M,G}. \tag{23}$$

Under 1-shot available-case averaging, MDGFM obtains the highest mean score (31.68). However, after coverage adjustment, MDGPT becomes the strongest 1-shot model (31.14), because it is feasible on all datasets. Under 5-shot coverage-aware evaluation, GCOPE becomes the strongest model (45.20), slightly ahead of MDGFM's coverage-aware score (40.20). This reversal shows that reporting only available-case performance overestimates specialized or memory-intensive GFMs. Feasibility is therefore not an implementation artifact; it is part of the model's practical transfer behavior.

Observation G.3: GFM progress should be measured by accuracy, elasticity, and coverage jointly. The GFM analysis based on Table 11 suggests that a single few-shot score is insufficient. A strong GFM should not only achieve high accuracy when it runs, but should also improve with additional labels and remain feasible across structurally difficult regimes. In this sense, GCOPE is a robust generalist, MDGFM is a high-performing but less coverage-stable model, and GraphGluing is a geometry-sensitive specialist. This provides a more nuanced evaluation than a flat GFM leaderboard, where OOM cases and regime-specific label elasticity are hidden.

Table 12: Macro-F1 results on table Node Classification (NC) task.

Baselines	Carcinogenesis	Hepatitis	PTE	Toxicology	F1
GCN	46.36±1.90	81.89±0.42	79.63±1.79	**41.14±9.10**	0.48±0.27
GAT	56.81±4.51	77.74±1.40	78.30±3.12	44.65±0.87	0.22±0.01
GraphSAGE	**64.31±1.18**	**80.59±1.24**	81.38±0.08	**49.70±3.74**	0.23±0.05
MLP	48.87±2.40	66.98±4.50	78.98±0.90	35.51±0.00	0.63±0.27
PCNet	**52.82±2.02**	**82.86±2.13**	**80.47±0.02**	35.94±1.25	0.20±0.01
HGNN	47.71±5.58	60.91±0.74	76.76±1.39	34.77±0.91	0.73±0.21
HAT	30.19±0.51	37.63±1.16	**85.62±3.02**	39.64±4.31	**2.74±0.29**
HGCNN	47.14±2.19	42.92±1.90	62.58±4.65	35.90±3.05	0.79±0.26
HyboNet	43.61±1.77	64.50±4.03	30.23±0.00	31.00±0.00	**1.20±0.16**
CUSP	43.92±8.79	76.06±0.51	44.16±11.13	36.93±1.81	0.75±0.21
QGCN	49.29±9.77	58.88±10.13	36.90±1.81	39.33±2.64	1.26±0.60
GraphMoRE	54.06±4.79	79.67±2.05	78.29±1.06	36.09±1.22	0.74±0.12

Specialist–robustness behavior on table-derived graphs. Finally, we examine table-derived NC results in Table 12 through rank mean and rank volatility. For each model, we compute the mean rank and standard deviation of ranks across the five table-derived datasets. GraphSAGE has the best mean rank (3.4), wins two datasets, and appears in the top-3 on four out of five datasets. This makes it the strongest robust baseline on table-derived graphs. In contrast, HAT has the highest rank volatility (5.61): it wins PTE and F1, but ranks last on Carcinogenesis and Hepatitis. PCNet is another specialist, winning Hepatitis but ranking last on F1. GraphMoRE has a moderate mean rank (5.4) but low rank volatility (1.82), indicating stable but rarely dominant behavior.

When restricting Table 12 to the four medical table-derived datasets, the conclusion becomes even sharper. GraphSAGE achieves the best medical mean performance (68.99) and the best average medical rank (1.75), while HAT drops to an average medical rank of 7.25 despite winning PTE. Therefore, HAT's strong overall impression is heavily driven by dataset-specific wins rather than broad robustness. This is consistent with the curvature-skewness view: table-derived graphs can contain strong tail signals, but different schema-induced tails favor different inductive biases.

Observation G.4: Table-derived graphs require reporting both winner counts and rank volatility. The results in Table 12 show that the model with the most striking wins is not necessarily the most reliable model. HAT demonstrates high upside but poor stability, while GraphSAGE provides the most consistent performance across medical table-derived datasets. This supports the central claim of CURVBENCH from a different angle: benchmark conclusions should not be reduced to a single averaged score. For heterogeneous relational data, robust evaluation must distinguish between generalists, specialists, and methods whose success is concentrated in narrow structural conditions.

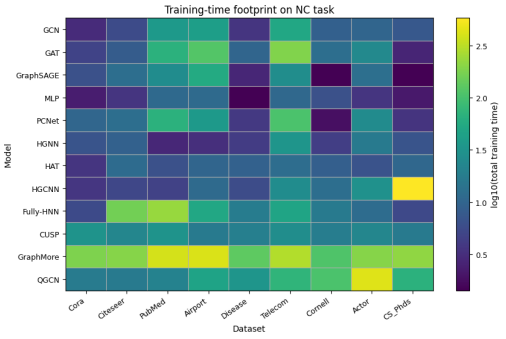


Figure 2: Total training time heatmap on Node Classification (NC) task.

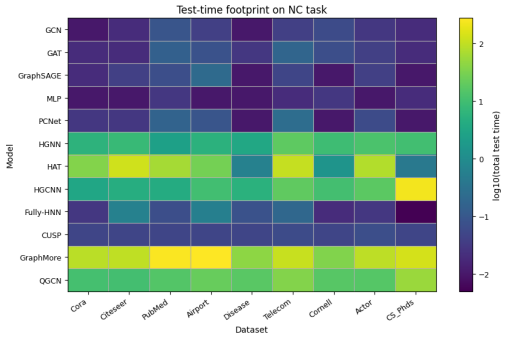


Figure 3: Toatl test time heatmap on Node Classification (NC) task.

H EFFICIENCY ANALYSIS

We further analyze the computational efficiency of different model families on the NC task. As shown in figure 2 and figure 3, we visualize the runtime footprint through log-scale heatmaps and a train–test efficiency landscape. Before visualization, we correct two apparent logging outliers: the training time of HyboNet on PubMed is adjusted from 2278.29 to 227.83, and the training time of QGCN on Cornell is adjusted from 10472.98 to 104.73. These two entries are several orders of magnitude larger than neighboring runs and would otherwise dominate the visualization.

The training-time heatmap shows that Euclidean and feature-only methods are generally the most efficient. MLP has the lowest median training time (3.59), followed by HGNN (6.71) and GCN (9.05). Among standard message-passing baselines, GCN, GraphSAGE, GAT, and PCNet remain within a similar order of magnitude, with median training times between 9.05 and 12.53. In contrast, GraphMoRE is substantially more expensive, with a median training time of 197.20, reflecting the overhead of adaptive Riemannian experts. QGCN also incurs a high median training cost (46.22), consistent with the additional computation required by pseudo-Riemannian geometry.

Figure 4 reveals an even sharper separation. MLP, GCN, GAT, GraphSAGE, PCNet, HyboNet, and CUSP all remain below 0.1 median test time, indicating that they are efficient at inference once trained. However, several non-Euclidean models introduce substantial inference overhead. HGNN, HGCNN, QGCN, HAT, and GraphMoRE have median test times of 7.92, 9.97, 16.13, 37.12, and 97.07, respectively. This suggests that the computational burden of non-Euclidean modeling is not limited to optimization; for some architectures, it persists during inference due to manifold operations, geometry-specific transformations, or expert routing.

The train–test efficiency landscape further separates models into three groups. First, MLP and standard Euclidean GNNs occupy the low-cost region, making them strong efficiency baselines. Second, CUSP and HyboNet have moderate training cost but low test-time cost, suggesting that some geometric or spectral overhead can be amortized after training. Third, QGCN, HAT, HGCNN, and GraphMoRE are high-overhead methods, especially at inference. GraphMoRE is the most expensive model overall, combining the largest median training time with the largest median test time. Therefore, efficiency should be considered jointly with performance: geometry-aware models may improve robustness or regime alignment, but their practical value depends on whether the performance gain justifies the additional train-time and inference-time cost.

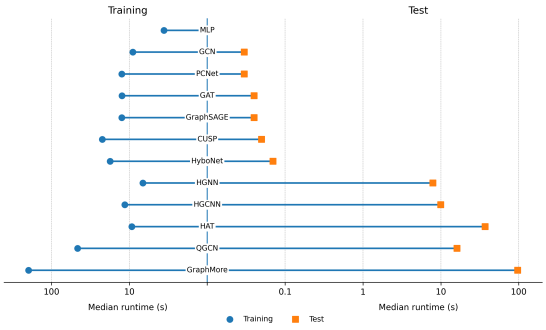


Figure 4: Mirrored efficiency diagram across models.

Observation H: Geometry-aware modeling introduces distinct efficiency profiles. The efficiency results show that computational cost is itself geometry-dependent. Euclidean baselines are consistently lightweight, while adaptive or manifold-heavy methods often incur substantial overhead. Importantly, training cost and inference cost do not always move together. For example, HyboNet has nontrivial training cost but remains relatively cheap at inference, whereas HAT and GraphMoRE are expensive during testing. This distinction matters for deployment: a method that is acceptable for offline training may still be impractical for repeated inference. CURVBENCH evaluates not only whether a model is accurate under a curvature regime, but also whether its geometric inductive bias is computationally affordable.