Towards Geometry-Grounded Layer-Wise Adaptivity for Graph Foundation Models

Shruti Sarika Chakraborty [1] **Peter Minary** [1]

Abstract

As the receptive field of a graph neural network expands across hops, the structural bottlenecks it encounters change, suggesting that different message-passing layers may favour different operators. Yet heterogeneous per-layer architectures produced by NAS are typically treated as optimisation artefacts, while GNNs, graph transformers, and graph foundation models (GFMs) largely retain homogeneous backbones that repeat one operator across depth. Using NAS as a diagnostic, we test (1) whether the per-layer signal is real rather than optimisation noise, (2) robust across architectural backbones and operator sets, and (3) responsive to graph geometry. We develop an ensemble-consensus protocol over independent search seeds and evaluate 31 configurations spanning three architectural envelopes and two operator spaces. Three complementary statistical tests, brute-force evaluation where tractable, and random-ensemble controls place per-layer preferences far above chance; the resulting consensus architectures are heterogeneous and match or approach SOTA on six of eight benchmarks. For geometric grounding, controlled synthetic experiments vary only graph geometry, while operators differ either in their aggregation step alone or as full GNN operators. Operator rankings and performance gaps shift across geometric regimes, with node-level preferences also tracking bottleneck position and prediction target. We thus elevate layer-wise heterogeneity from a recurring search output to a statistically validated, geometrically responsive phenomenon, opening the question of whether graph geometry admits a principled mapping to operator preference, with direct implications for how GFMs choose their backbones.

[1]Department of Computer Science, University of Oxford, Oxford, United Kingdom. Correspondence to: Shruti Sarika Chakraborty <shruti.chakraborty@cs.ox.ac.uk>.

The Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea.

1. Introduction

As the receptive field of a graph neural network expands across hops, the local graph structure it processes varies with depth. Topping et al. (2022) formalised this through Balanced Forman Curvature (BFC), an edge-level descriptor that quantifies how bottlenecks concentrate, and Giovanni et al. (2023) showed that graph topology, in particular nodes at high commute time, drives over-squashing in message-passing networks. *If the geometry a network must process changes with depth* – visible within a single MalNet-Tiny graph in Figure 1, where the negative-curvature edge fraction shifts from 45% at hop 1 to 22% at hop 4, and across datasets in Figure 15 (Appendix R.4), then *a question naturally follows: should different message-passing layers favour different operators for optimal information flow, or is the standard practice of repeating one operator at every depth structurally appropriate?*

The neural architecture search literature consistently produces heterogeneous per-layer architectures (Figure 19) across differentiable, reinforcement-learning, evolutionary, and recent LLM-driven approaches (Zhao et al., 2021; Qin et al., 2021; Wei et al., 2021; Wang et al., 2025c; Qin et al., 2022a; Gao et al., 2019; Lai et al., 2020; Liu & Liu, 2023; Shi et al., 2022; He et al., 2025a; Sasaki, 2025; Wang et al., 2025b), with NAS-Bench-Graph (Qin et al., 2022b) confirming this at scale. Yet layer-wise heterogeneity is reported as a search by-product, and whether the per-layer signal reflects a genuine structural property of graph data or an optimisation artefact remains open. The homogeneity assumption is most consequential for graph foundation models (GFMs) (Liu et al., 2024; He et al., 2025b; Wang et al., 2024; Galkin et al., 2024), which aim to generalise a fixed backbone across graphs of widely varying geometric profiles. A recent line questions this fixity along different axes - AutoGFM (Chen et al., 2025) adapts the architecture per input from a learned invariant signature, GraphAny (Zhao et al., 2025) mixes a fixed set of operators per node, GOBLIN (Gutteridge et al., 2026) conditions its operator mixture on the range a task demands, and AnyGraph (Xia & Huang, 2024) routes between expert backbones per domain. But the adaptation only chooses *which* operator or expert to use; the chosen operator is then repeated identically across the stack.

AutoGFM is the one case with a genuinely heterogeneous per-layer stack, leaving open the question of whether that layer-wise variation is structurally grounded. A separate strand (Finkelshtein et al., 2025) pursues generality through equivariance rather than adaptivity, designing the right symmetries into the model instead of adapting to each input. The same homogeneity persists in modern Graph Transformers (Rampášek et al., 2022; Ying et al., 2021; Kreuzer et al., 2021; Wu et al., 2021), and GT-specific NAS studies (Zhang et al., 2023; Wang et al., 2025a; Song et al., 2025) either fix the per-layer GNN or vary it on a hybrid backbone without treating heterogeneity as a structural phenomenon in its own right. Across all of these, the granularity at which architectures should adapt, and the structural property on which adaptation should be conditioned, remain open. Establishing whether the per-layer signal is real, robust, and exhibits directional dependence on the graph geometry is a precondition for any geometry-aware choice of granularity. We defer a detailed literature treatment to Appendix B.

We use neural architecture search as a scientific probe and ask three questions in sequence.

Q1. Is the per-layer signal real, or optimisation noise?
Q2. Is it robust across different architectures and operator search spaces?
Q3. Does it vary systematically with graph geometry?

For **Q1**, we develop an ensemble-consensus protocol in which independent DARTS seeds vote on the operator at each layer. A hybrid rule takes the majority choice where seeds concentrate, and otherwise selects the operator whose architecture parameter, averaged across seeds, is largest. We validate the resulting consensus against optimisation noise through brute-force evaluation on tractable configurations, a random-ensemble control, and three complementary statistical tests, which reject chance-level operator selection in 20/31 configurations after Bonferroni correction.

For **Q2**, we embed the protocol in three model architectures, which we term envelopes, spanning the GT-to-GNN spectrum. *Entangled-GT* draws attention queries from node features and keys and values from a per-layer GNN. *DisEntangled-GT* runs the local and global streams fully separated. *Pure GNN* uses no attention at all. Each envelope is searched over two operator sets: $\mathbf{SS_1} = \{\text{GIN}, \text{GCN}, \text{GAT}, \text{SAGE}\}$ (non-edge-aware) and $\mathbf{SS_2} = \{\text{GINE}, \text{GatedGCN}, \text{GATv2}\}$ (edge-aware). Per-layer single-GNN selection emerged from prior exploration of finer granularities (Appendix Q). The per-layer signal is statistically significant across all three envelopes, and the resulting consensus architectures are heterogeneous, though the operator chosen at each layer varies by envelope.

For **Q3**, we isolate the geometry and aggregation relationship through three synthetic graphs spanning the BFC spectrum, from the positively curved clique ($+1.11$) through the grid ($\{+0.17, +0.33\}$) to the negatively curved tree ($\{-0.33, 0\}$). Across these graphs, operators compete in a message-passing prediction task, and we find that the winning operator varies across geometric regimes, as does the gap between operators' mean-squared errors (MSEs). A *decoupled* experiment varies only the aggregation step, and a *bundled* experiment varies the full GNN operator. The inter-operator spread grows monotonically from clique to grid to tree in the decoupled setting, whereas in the bundled setting it is non-monotonic, with the grid spread the smallest of the three. Each operator's stability across seeds also shifts with geometry, and the same pattern reappears within a single graph, where per-node preference tracks both the bottleneck position and the node carrying the prediction.

Together, these results support two conclusions. First, the layer-wise heterogeneity surfaced by NAS holds up under statistical testing rather than reflecting search noise. Second, operator preference responds to graph geometry, pointing to a structural property of graph data (Section 3).

Contributions. *(1) An ensemble-consensus protocol for GNAS as a probe.* A multi-seed differentiable search with hybrid voting and mean-fallback aggregation, designed to surface the per-layer operator-preference signal while addressing the seed-instability of differentiable NAS (Zela et al., 2020), with its consensus validated against brute-force evaluation and a random-ensemble control. *(2) Empirical and statistical validation of layer-wise heterogeneity.* Across 31 configurations spanning three envelopes and two operator search spaces, $20/31$ reject chance-level operator selection at the Bonferroni-corrected threshold. *(3) Controlled synthetic experiments linking geometry to operator preference.* On synthetic graphs that vary only in geometry, operators compete at a message-passing prediction task. The gap between their MSEs grows monotonically across geometric regimes when only the aggregation step varies, and non-monotonically when the full operator varies. The same pattern appears within a single graph: per-node operator preference responds to bottleneck position and to which node carries the prediction, with the gap reaching $5\times$ (decoupled) and $14\times$ (bundled) at the tree root. *(4) Theoretical grounding.* Two propositions and a remark establish that homogeneous architectures occupy a vanishing n^{1-L} fraction of the search space (Prop. 2.3), heterogeneous local minima outnumber homogeneous ones by $n^{L-1} - 1$ (Prop. 2.5), and functionally distinct envelopes induce distinct local-minima sets (Remark 2.6).

Taken together, these findings point to a per-layer view: the geometry a network processes changes with depth (Figure 1), and if operator preference follows geometry, it should change with depth too. Showing this directly on real graphs remains open.

2. Method

2.1. Preliminaries

Notation. For graph $G = (V, E)$, $H^{(\ell)} \in \mathbb{R}^{|V|\times d}$ is the node-embedding matrix entering layer ℓ, with $H^{(0)}$ the input features. The candidate operator pool is $\mathcal{O} = \{o_1, \dots, o_K\}$ with $K = |\mathcal{O}|$. Multi-head attention uses M heads with per-head projections $W^{(\ell)}_{Q,m}, W^{(\ell)}_{KV,m} \in \mathbb{R}^{d\times d_m}$ where $d_m = d/M$, and ϕ is a learnable projection to the query stream (Appendix O).

DARTS. An L-layer GNN updates as $H^{(\ell)} = o(H^{(\ell-1)}, E)$ with $o \in \mathcal{O}$. DARTS (Liu et al., 2019) relaxes this discrete choice into a mixed operator parameterised by architecture parameters $\boldsymbol{\alpha}^{(\ell)} \in \mathbb{R}^K$:

$$\bar{o}^{(\ell)}(H, E) = \sum_{o\in\mathcal{O}} \frac{\exp(\alpha^{(\ell)}_o)}{\sum_{o'} \exp(\alpha^{(\ell)}_{o'})} \cdot o(H, E). \tag{1}$$

After joint bi-level optimisation of $\boldsymbol{\alpha}$ and network weights w (60/40 train/val split, second-order variant; Appendix N), the discrete operator at layer ℓ is $\pi^\star_\ell = \arg\max_o \alpha^{(\ell)}_o$.

Entangled-GT and asymmetric attention. We adopt the per-layer asymmetric Graph Transformer from our earlier study (Chakraborty & Minary, 2025) where queries come from node features through ϕ, and keys and values come from a per-layer GNN operator $o^{(\ell)}$:

$$\begin{aligned} Q^{(\ell)}_m &= \phi(H^{(\ell)})\, W^{(\ell)}_{Q,m}, \\ K^{(\ell)}_m &= V^{(\ell)}_m = o^{(\ell)}(H^{(\ell)}, E)\, W^{(\ell)}_{KV,m}. \end{aligned} \tag{2}$$

This preserves the feature–structure asymmetry of GMT (Baek et al., 2021) while applying a single GNN per layer, reducing GNN forward cost from $\mathcal{O}(2M \cdot \mathrm{cost}_{\mathrm{GNN}})$ to $\mathcal{O}(\mathrm{cost}_{\mathrm{GNN}})$. Per-layer single-GNN selection emerged from prior exploration of alternative granularities (per-head, per-group, MoE; Appendix Q). During search, $o^{(\ell)}$ in Equation 2 is replaced by $\bar{o}^{(\ell)}$ from Equation 1; after discretisation, $\pi^\star_\ell$ takes its place.

2.2. Layer-Wise Architecture Search Setup

A layer-wise architecture is an ordered sequence $\pi = (\pi_1, \dots, \pi_L) \in \mathcal{S} = \mathcal{O}^L$, with layer ℓ updating embeddings as $H^{(\ell)} = \pi_\ell(H^{(\ell-1)}, E)$. The search problem is

$$\pi^\star = \arg\min_{\pi\in\mathcal{S}} \mathcal{L}_{\mathrm{val}}(w^\star(\pi), \pi), \tag{3}$$

$$w^\star(\pi) = \arg\min_{w} \mathcal{L}_{\mathrm{train}}(w, \pi). \tag{4}$$

The formulation is backbone-agnostic; we instantiate it on Entangled-GT and two further envelopes (Section 2.5, Figure 12).

2.3. Ensemble Consensus and the Hybrid Rule

A single DARTS run is unreliable: bi-level optimisation is non-convex, and DARTS is sensitive to initialisation (Zela et al., 2020). We run DARTS independently with N seeds. For run m at layer ℓ, the per-run winner is $\pi_{m,\ell} = \arg\max_o \alpha^{(\ell)}_{m,o}$. [1] For each operator $o_k \in \mathcal{O}$ with $k \in \{1, \dots, K\}$, we summarise the runs by the per-layer vote count $c_{\ell,k} = \sum_{m=1}^N \mathbf{1}[\pi_{m,\ell} = o_k]$ and mean architecture parameter $\bar{\alpha}^{(\ell)}_o = \frac{1}{N}\sum_{m=1}^N \alpha^{(\ell)}_{m,o}$.

Hybrid rule. If a clear majority of seeds agree at a layer, we take the vote winner; otherwise, we fall back to the highest mean architecture parameter (illustrated in Figure 13):

$$\pi^{\mathrm{cons}}_\ell = \begin{cases} \arg\max_o\, c_{\ell,o}, & \text{if } \max_o c_{\ell,o} \geq 0.6\,N, \\ \arg\max_o\, \bar{\alpha}^{(\ell)}_o, & \text{otherwise.} \end{cases} \tag{5}$$

The threshold $0.6N$ requires six of ten seeds before voting resolves the layer. Per-layer confidence is $\kappa_\ell = \max_o c_{\ell,o}/N$; layer-dependence of $c_{\ell,\cdot}$ is tested in Section 2.6.

2.4. Geometric Descriptor: Hop-Stratified Over-Squashing Pressure

To examine whether per-layer operator preferences track depth-varying geometric structure on real graphs, we define a hop-stratified geometric descriptor based on Balanced Forman Curvature (BFC) (Topping et al., 2022) ($\mathrm{Ric}(e)$ and full algorithm in Appendix R). For source $v_0 \in V$ and hop ℓ, the hop ring $\Gamma_\ell(v_0) \subseteq E$ collects edges (a, b) with $\max(\mathrm{dist}_G(v_0, a), \mathrm{dist}_G(v_0, b)) = \ell$, where dist_G is the shortest-path (BFS) distance in G (illustrated in Figure 14, Appendix R) and the over-squashing pressure at hop ℓ sums the negatively curved edges in that ring:

$$S^{(\ell)}_{v_0} = \sum_{\substack{e\in\Gamma_\ell(v_0)\\ \mathrm{Ric}(e)<0}} \mathrm{Ric}(e), \qquad \bar{S}^{(\ell)} = \frac{1}{|V|}\sum_v S^{(\ell)}_v. \tag{6}$$

We stratify pressure *marginally* by hop: each ring $\Gamma_\ell(v_0)$ contains only edges whose farther endpoint is at distance exactly ℓ, so $S^{(\ell)}$ isolates the bottleneck pressure newly encountered at hop ℓ rather than accumulated over hops $\leq \ell$. This marginal construction is what allows us to ask whether the geometry processed *at* a given layer differs across depths; a cumulative variant would conflate each layer with all shallower layers. The ring is computed separately for each source v_0, and we aggregate over all sources. A single edge, therefore, appears once per source

[1] We reuse m as the run index here; m denotes attention heads only in the multi-head attention and Transformer equations, and the two scopes never co-occur.

that places it in ring ℓ, so the per-hop counts in Figure 1 tally edge–source pairs, not distinct edges: the same edge can be counted at different hops when viewed from different sources. Dataset-level statistics aggregate $\bar{S}^{(\ell)}$ across graphs. The descriptor confirms that real graphs occupy distinct depth-varying geometric regimes (CLUSTER vs MalNet-Tiny in Appendix R.4) — a precondition for any per-layer geometry-to-operator mapping. The construction extends to other geometric descriptors (effective resistance, spectral gap). A single-graph illustration is shown in Figure 1: within one MalNet-Tiny graph, the fraction of negatively curved edges shifts substantially across hops, confirming that the geometric regime processed at layer ℓ differs from that at $\ell \pm 1$ even when the graph is held fixed.

2.5. Architectural Backbones and Search Spaces

Beyond Entangled-GT (Equation 2), we instantiate two further envelopes (Figure 12). **DisEntangled-GT**: two parallel streams flow through each layer, with attention stream supplying queries $Q_m^{(\ell)} = \phi(H_{\text{attn}}^{(\ell)})W_{Q,m}^{(\ell)}$ and GNN stream supplying keys and values $K_m^{(\ell)} = V_m^{(\ell)} = \pi_\ell(H_{\text{gnn}}^{(\ell)}, E)W_{KV,m}^{(\ell)}$, with the residual on the attention stream only. **Pure GNN**: stacked GNN layers with no attention, $H^{(\ell+1)} = \pi_\ell(H^{(\ell)}, E)$. We use two operator sets, $\mathbf{SS_1} = \{\text{GIN}, \text{GCN}, \text{GAT}, \text{SAGE}\}$ (non-edge-aware) and $\mathbf{SS_2} = \{\text{GINE}, \text{GatedGCN}, \text{GATv2}\}$ (edge-aware); operator equations are in Appendix S. The three architectural envelopes (Entangled-GT, DisEntangled-GT, Pure GNN) are illustrated in Figure 12 (Appendix K). The rationale for choosing precisely these three envelopes, rather than, for example, the GPS parallel-path design, is detailed in Appendix K. The three envelopes span a controlled local-global mixing spectrum, and the aim of this study is not to propose a new NAS method or GT architecture but to test whether layer-wise heterogeneity holds across functionally distinct envelopes.

2.6. Statistical Framework for Q1

We formalise Q1 as a joint-rejection criterion over a family of complementary statistical tests applied to the per-layer vote counts $c_{\ell,k}$ from the ensemble (Section 2.3). Let $\mathcal{C} \subseteq \mathcal{D} \times \mathcal{SS} \times \mathcal{B}$ denote the set of (dataset, search space, envelope) configurations evaluated, with $|\mathcal{C}| = N_c$; each $c = (d, s, b) \in \mathcal{C}$ has L_c layers and $L_{\text{total}} = \sum_c L_c$. Let H_0 denote the null that DARTS picks operators randomly with $p_0 = 1/K$, so $(c_{\ell,1}, \ldots, c_{\ell,K}) \sim \text{Multinomial}(N, p_0)$ with $p_0 = (1/K, \ldots, 1/K)$. A test T on $(c_{\ell,1}, \ldots, c_{\ell,K})$ probes an *aspect* $\mathcal{A}_T$, the set of departures from H_0 that T is sensitive to.

Definition 2.1 (Complementary family of tests)**.** A family $\mathcal{T} = \{T_1, \ldots, T_J\}$ of tests on $(c_{\ell,1}, \ldots, c_{\ell,K})$ is *complementary* when the aspects $\mathcal{A}_{T_i}$ are pairwise non-containing: $\mathcal{A}_{T_i} \not\subseteq \mathcal{A}_{T_j}$ and $\mathcal{A}_{T_j} \not\subseteq \mathcal{A}_{T_i}$ for all $i \neq j$.

Definition 2.2 (Joint-rejection criterion)**.** For each $T_j \in \mathcal{T}$, let $p_j^{(c,\ell)}$ be the within-configuration Bonferroni-corrected p-value (multiplier L_c), and $R_j = \{(c, \ell) : p_j^{(c,\ell)} < \alpha\}$. Under H_0, $\mathbb{E}_{H_0}[|R_j|] \leq \alpha \cdot N_c$ (Appendix I). The per-layer signal is declared real when $|R_j| \gg \alpha \cdot N_c$ for all $j = 1, \ldots, J$.

Instantiation. We instantiate $\mathcal{T}$ with $J = 3$ complementary tests: T_1 chi-square goodness-of-fit on $(c_{\ell,1}, \ldots, c_{\ell,K})$ (aspect: deviation in the full vote distribution), T_2 exact binomial with post-hoc winner correction (aspect: concentration on a single operator), and T_3 Friedman test on $\alpha_{i,\ell,k}$ (seed i, layer ℓ, operator k) across seeds (aspect: rank consistency in continuous strengths). Aspects are pairwise non-containing per Definition 2.1; full definitions are in Appendix I. At higher seed counts ($N = 38$ runs per Peptides task, with a supplementary pooled $N = 76$ analysis across both tasks), a chi-square test of independence between layer position and operator choice (Appendix J) directly tests layer-dependence of operator preference, complementing the per-layer signal-reality tests above.

Configuration-level pooling. For a coarser per-configuration view, we pool per-layer chi-square p-values via Fisher's method and apply across-configuration Bonferroni at α/N_c; the count of configurations rejecting at this threshold is reported in Table 1.

2.7. Counting: Heterogeneity is the Natural State of the Search Space

Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ be the operator pool with $n = |\mathcal{O}| = K$ and L layers. The architecture search space is $\mathcal{S} = \mathcal{O}^L$ with $|\mathcal{S}| = n^L$, the homogeneous subset is the diagonal $\mathcal{H}_{\text{homo}} = \{(o, \ldots, o) : o \in \mathcal{O}\}$, and the heterogeneous complement is $\mathcal{H}_{\text{het}} = \mathcal{S} \setminus \mathcal{H}_{\text{homo}}$.

Proposition 2.3 (Measure of Homogeneity)**.** $|\mathcal{H}_{\text{homo}}|/|\mathcal{S}| = n/n^L = n^{1-L}$.

For $n = 4, L = 6$, this fraction is $1/1024$. Restricting architecture search or model design to the homogeneous diagonal — the standard practice in fixed-backbone graph transformer models such as GraphGPS and Graphormer, in graph foundation models, and in most NAS-Bench-Graph baselines — restricts to a measure-vanishing corner of the search space. Heterogeneity is the natural state. The burden of justification rests on *why* homogeneity should be preferred. Proof in Appendix D.

2.8. Local Minima: Search Outcomes Concentrate on Heterogeneous Architectures

We now show that the same conclusion holds when the search converges to a *local minimum* of the loss surface —

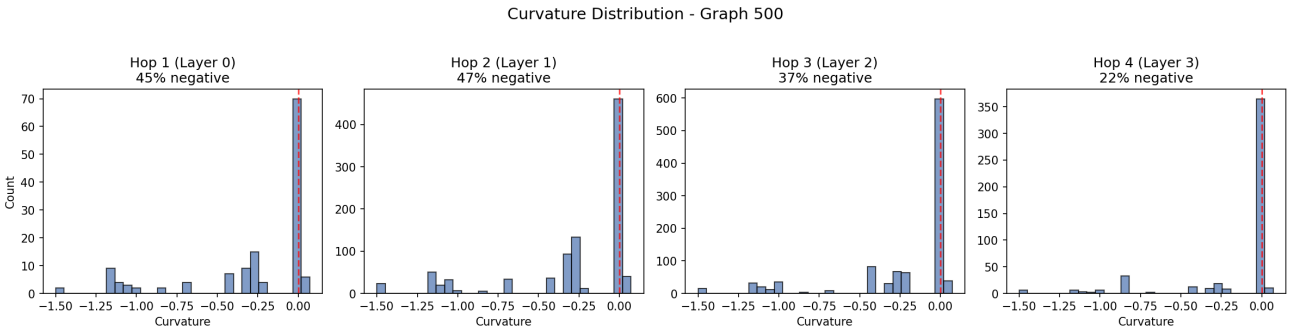


Figure 1. **Per-hop curvature distribution for a single MALNET graph.** Edge-level Balanced Forman Curvature within each hop ring $\Gamma_\ell(v_0)$, aggregated over all sources v_0. Each edge is placed in the ring matching its farther endpoint's distance, computed per source and then summed over sources, so the counts are edge–source pairs: the same edge can fall at different hops for different sources. The negative-curvature fraction shifts with depth ($45\%, 47\%, 37\%, 22\%$ at hops 1–4), with deeper hops concentrating near zero and shallower hops keeping a negative tail. Raw counts rise, then fall as the BFS frontier expands, eventually saturating in a finite graph. Dataset-level aggregation in Figure 15 (Appendix R.4).

the actual outcome of every NAS algorithm.

Setup. Equip $\mathcal{S} = \mathcal{O}^L$ with the one-layer-flip neighbourhood structure: $\pi, \pi' \in \mathcal{S}$ are neighbours if they differ in exactly one layer. Each π has $|\mathcal{N}(\pi)| = L(n-1)$ neighbours, and the closed neighbourhood is $\overline{\mathcal{N}}(\pi) = \{\pi\} \cup \mathcal{N}(\pi)$ with size $L(n-1)+1$. Let $\mathcal{L} : \mathcal{S} \to \mathbb{R}$ denote validation loss; π is a local minimum if $\mathcal{L}(\pi) \leq \mathcal{L}(\pi')$ for all $\pi' \in \mathcal{N}(\pi)$.

Assumption 2.4 (Genericity)**.** The loss values $\{\mathcal{L}(\pi) : \pi \in \mathcal{S}\}$ are pairwise distinct (which holds in practice as continuous random variables coincide with probability zero) and exchangeable: any permutation of values across architectures yields the same joint distribution. Assumption 2.4 is an idealisation. We show in the Appendix E (Remarks E.1 - E.3) that the conclusion is robust to flat regions and correlated losses, where the counting result of Proposition 2.3 takes over even when distinctness fails.

Proposition 2.5 (Local Minima Distribution)**.** *Under Assumption 2.4,*

$$\frac{\mathbb{E}[\#\{\textit{heterogeneous local minima}\}]}{\mathbb{E}[\#\{\textit{homogeneous local minima}\}]} = n^{L-1} - 1. \quad (7)$$

For $n = 4, L = 6$, this ratio is 1023. Search procedures — DARTS, evolutionary search, RL, local search converge to local minima, and Proposition 2.5 shows that under generic conditions local minima are dominantly heterogeneous, with dominance growing exponentially in L. This explains, structurally, why search consistently produces heterogeneous architectures regardless of the method. Figure 6 (Appendix E) shows the empirical realisation: across five brute-force-evaluated configurations, homogeneous architectures rarely place in the top tier as L grows, while DARTS consensus architectures consistently do on MalNet-Tiny ($L = 4$, $n = 81$ candidates), homogeneous architectures spread from rank 9 to rank 74 on validation while DARTS consensus lands at rank 29. Proof and robustness to flat or correlated landscapes are in Appendix E.

2.9. Envelopes Induce Distinct Local Minima Sets

We now examine what changes when the loss function itself is modulated by the architectural envelope that wraps the per-layer GNN sequence.

Setup. An *architectural envelope* is a pair $\mathcal{E} = (\pi, B)$ where $\pi \in \mathcal{O}^L$ is the per-layer GNN sequence and B is a wrapper fixing how it composes with the rest of the model: $B = \emptyset$ (Pure GNN), $B = (\text{attn}, \text{FFN}, \text{residual})$ with the GNN supplying keys and values to attention (Entangled-GT), or its parallel-stream variant (DisEntangled-GT). Let $\text{LM}(B) = \{\pi \in \mathcal{O}^L : \pi \text{ is a local minimum of } \mathcal{L}_B\}$ under the one-layer-flip structure of Proposition 2.5, and let $\mathcal{I}_\ell^B$ be the distribution over inputs the per-layer GNN at layer ℓ receives under B. The *functional envelope divergence* is $d_{\text{env}}(B_A, B_B) = \sum_{\ell=1}^{L} \text{KL}(\mathcal{I}_\ell^{B_A} \| \mathcal{I}_\ell^{B_B})$. The two wrappers are *functionally distinct* if $d_{\text{env}}(B_A, B_B) > 0$.

Remark 2.6 (Envelopes Induce Distinct Local Minima). When two wrappers B_A, B_B are functionally distinct ($d_{\text{env}}(B_A, B_B) > 0$), the per-layer GNN at some layer $\ell^\star$ receives input from a different distribution under B_A versus B_B, inducing different gradients with respect to the operator at $\ell^\star$. Generically, this yields different orderings of $\mathcal{L}_B$ across architectures within affected closed neighbourhoods, so $\text{LM}(B_A) \neq \text{LM}(B_B)$ in general — with positive probability there exists $\pi \in \mathcal{O}^L$ that is a local minimum under one envelope but not the other. Empirical realisation in Figure 2 (SS2) and Figure 7 (SS1, Appendix F): the three envelopes disagree on consensus operators at multiple layers within the same (dataset, search space) configuration.

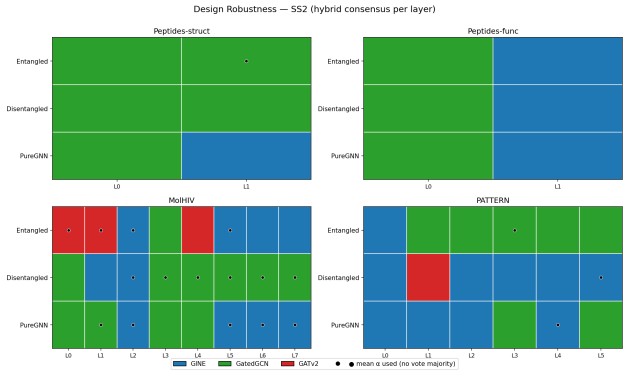


Figure 2. Empirical anchor: per-layer consensus operator across three envelopes (Entangled-GT, DisEntangled-GT, Pure GNN) on four SS2 datasets. Black dots mark layers where the hybrid rule (Equation 5) fell back to mean architecture parameters. The envelopes disagree on consensus operators at multiple layers within the same (dataset, search space) configuration, empirically realising Remark 2.6; SS1 in Figure 7.

2.10. Synthetic Experiments: Geometry and Aggregation

To test whether geometry alone drives aggregation preference, we construct minimal synthetic graphs in which only geometry varies (Figure 4, inspired by BuNN (Bamberger et al., 2025a)). Three regimes $G \in \{G_{\text{clique}}, G_{\text{grid}}, G_{\text{tree}}\}$ span the BFC (Topping et al., 2022) spectrum, with $\text{BFC}(G_{\text{clique}}) = +1.11$ (1 hop), $\text{BFC}(G_{\text{grid}}) \in \{+0.17, +0.33\}$ (2 hops), and $\text{BFC}(G_{\text{tree}}) \in \{-0.33, 0\}$ (2 hops). Nodes are partitioned into types; each predicts the mean feature value of a designated target group via message-passing, with 1D scalar features drawn from disjoint Gaussians per type to avoid feature-dimensionality confounds.

We run two experiments. The *decoupled* experiment fixes message $\psi(\mathbf{h}_j) = \mathbf{W}\mathbf{h}_j$ and update $\varphi = \text{MLP}$, varying only $\Sigma \in \mathcal{A}_{\text{dec}} = \{\text{sum}, \text{mean}, \text{degree-norm}, \max\}$; the *bundled* experiment varies the full GNN over $\mathcal{A}_{\text{bun}} = \{\text{GIN}, \text{GCN}, \text{GAT}, \text{SAGE-mean}, \text{SAGE-max}\}$. With $K = |\mathcal{A}|$, denoting the number of candidate operators, and operator indices in $\{1, \ldots, K\}$, we report for each regime G and operator k, the per-operator MSE $\mu_k(G) = \mathbb{E}[\text{MSE}_k(G)]$, inter-operator spread $\Delta(G) = \max_{i,j \in \{1,\ldots,K\}} |\mu_i(G) - \mu_j(G)|$, and best operator $k^\star(G) = \arg\min_k \mu_k(G)$. For two distinct regimes $G_1, G_2 \in \{G_{\text{clique}}, G_{\text{grid}}, G_{\text{tree}}\}$ with $G_1 \neq G_2$, we ask: $\Delta(G_1) \neq \Delta(G_2)$? and $k^\star(G_1) \neq k^\star(G_2)$? Decoupled-setting overall results and per-node breakdowns on the binary tree are demonstrated in Figure 3; bundled-setting results in Appendix G. Per-operator MSE, across-seed stability, inter-operator spread, and the winning operator for both settings across all three regimes are summarised in Table 3. Table 6 gives the formulations of operators from decoupled and bundled sets and the whole experimental setup is demonstrated in Figure 4.

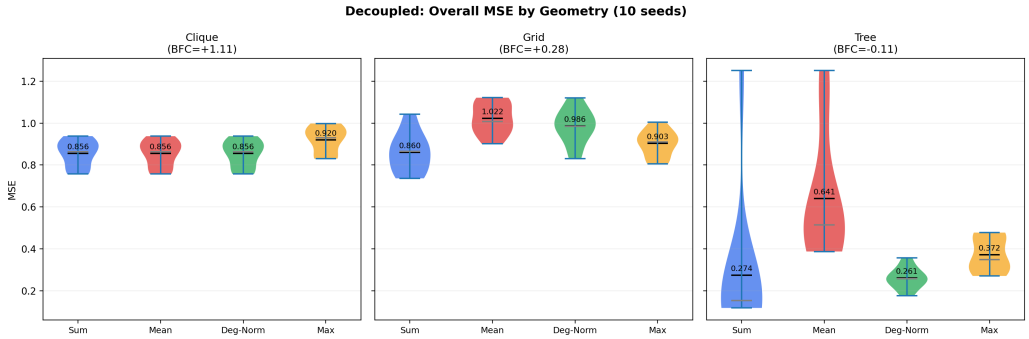
We additionally probe task dependence by varying the node that carries the prediction target in the binary tree. In the original assignment, the internal node I predicts the root R; we compare against a reversed assignment (I predicts a leaf L) and a neutral assignment (I carries no task). The three configurations share geometry and feature distributions but differ in which task pressure each intermediate node carries. Per-node MSE results across the three conditions are presented in Appendix H, and a similar experiment was conducted on the grid, too, in Figure 11.

3. Results

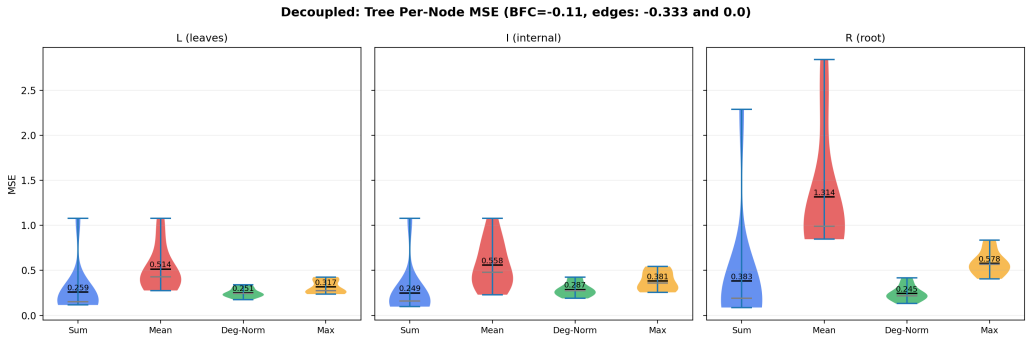
We evaluate across eight benchmarks from (Rampášek et al., 2022) spanning molecular property prediction, peptide property prediction, node classification, and malware call-graph classification (full statistics, splits, and hyperparameters in Appendix L). For controlled comparison with baselines (Dwivedi & Bresson, 2021) and GPS, we adopt the GPS (Rampášek et al., 2022) configuration (positional encoding, hidden dimensions, heads, dropout, pre/post-processing identical), adjusting only layer count to maintain a comparable parameter budget. The results are organised around the three questions of Section 2: **Q1** (*is the per-layer signal real?*), **Q2** (*is it robust across architectural envelopes and search spaces?*), and **Q3** (*does it vary systematically with graph geometry?*).

Q1: The per-layer signal is real. For every (dataset, search space, backbone) configuration, we test whether the per-layer operator votes show a real preference, against the null hypothesis that DARTS selects operators uniformly at random (Section 2.6). Across 31 configurations, $20/31$ reject the null at the Bonferroni-corrected threshold $\alpha/31 \approx 1.6 \times 10^{-3}$ and $24/31$ at nominal $\alpha = 0.05$. At the per-layer level, the three complementary tests yield observed rejection counts 30–40 times the Bonferroni null upper bound of ≤ 1.55 false-positive layers per test (Table 1). The strongest single configuration is CLUSTER (SS1, Entangled-GT) at $p = 1.49 \times 10^{-63}$. Per-configuration breakdowns are in Table 7 (Appendix I). Furthermore, a higher-N Peptides study ($N = 38$ runs per task; pooled $N = 76$ across Peptides-func and Peptides-struct) confirms the signal strengthens ($p = 2.98 \times 10^{-14}$ and 3.52×10^{-4} per task; 6.69×10^{-17} pooled) and is invariant to first-order vs second-order DARTS (Appendix J).

While the per-layer tests establish non-chance *preference* rather than heterogeneity directly, the resulting consensus architectures are empirically heterogeneous: Figures 2 and 7 show different operators selected at different layers within the same configuration, so the validated per-layer signal



(a) Across regimes.



(b) Per-node on G_{tree}.

Figure 3. Decoupled-setting MSE. **Top**: per-operator MSE across three geometric regimes; spread is minimal on G_{clique}, modest on G_{grid}, largest on G_{tree}. **Bottom**: per-node distribution on G_{tree}; root shows up to $5\times$ MSE differences.

manifests as genuine layer-wise heterogeneity rather than a uniform preference for one operator.

Q1 (continued): DARTS finds the right architectures. A genuine per-layer signal is only useful if DARTS converges to architectures that are actually good. We exhaustively evaluate all candidates in five small-enough settings and compare DARTS to a random-ensemble baseline that replaces DARTS with random per-layer sampling under the same rule (Table 2). DARTS consistently lands at or near the top, while the random ensemble is significantly worse: the MalNet-Tiny SS2 row is most decisive — DARTS reaches $26/81$ test, $29/81$ val while random falls to $64/81$ and $76/81$, with Fisher p-values separated by seven orders of magnitude (4.09×10^{-8} vs 0.62). Full ranking plots in Appendix T; a $N = 20$ robustness check on MalNet-Tiny (consensus identical to $N = 10$) is in Appendix U.

Q2: The signal is robust across envelopes and search spaces, but the operator identity is dataset-specific. Method Section 2.9 (Figure 2) already showed that heterogeneity persists across all three architectural envelopes (Entangled-GT, DisEntangled-GT, Pure GNN), while the specific operator at each layer differs across envelopes — illustrating Remark 2.6. The remaining question is whether heterogeneity itself is dataset-driven: do different datasets prefer different layer-wise architectures, or does one global pattern dominate? Figure 5 clusters the SS2 hybrid-consensus architectures hierarchically by per-layer fingerprint. Datasets are grouped by consensus architecture rather than domain label —for instance, MalNet-Tiny clusters with the molecular benchmarks. The grey bar reports per-layer confidence κ_ℓ, flagging low-confidence rows (e.g. MolHIV). Layer-wise heatmaps that make the per-layer pattern explicit are in Appendix V.

Table 1. Headline statistical evidence against chance-level operator selection. Config-level Fisher pools per-layer chi-square evidence; per-layer counts aggregate across all 166 layers. Null entries are Bonferroni false-rejection upper bounds: ≤ 0.05 for the across-config Bonferroni threshold (α/N_c), and ≤ 1.55 for the nominal $\alpha = 0.05$ and within-config per-layer thresholds.

Test	Observed	Null
Config-level (of 31)		
Fisher (Bonf.)	20/31	≤ 0.05
Fisher ($\alpha = 0.05$)	24/31	≤ 1.55
Per-layer (of 166)		
χ^2 goodness-of-fit	56/166	≤ 1.55
Exact binomial	51/166	≤ 1.55
Friedman	63/166	≤ 1.55

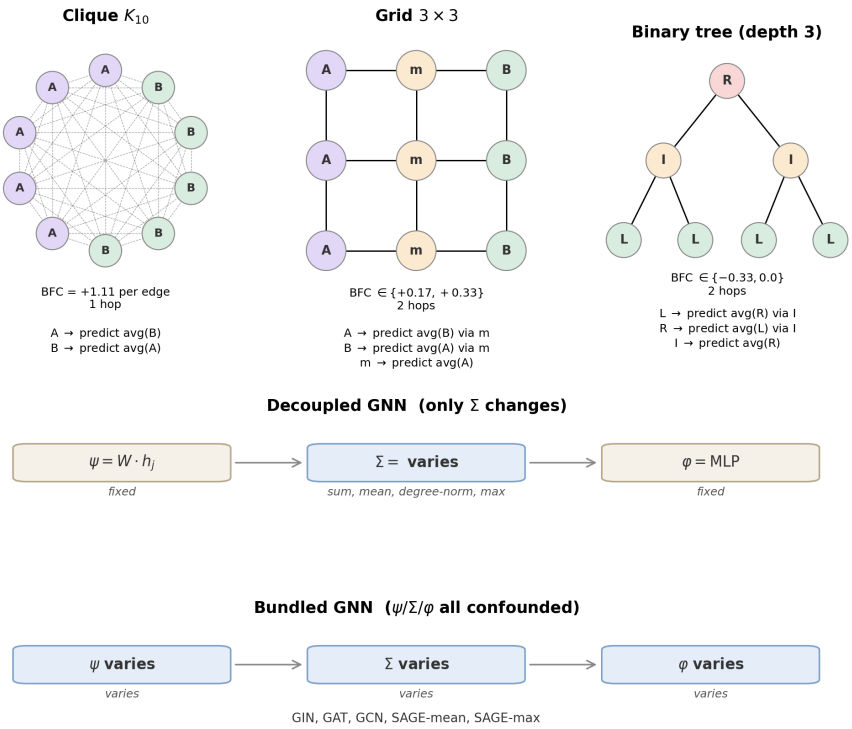


Figure 4. Synthetic graph regimes spanning $\mathrm{BFC} > 0$ (clique) to $\mathrm{BFC} < 0$ (tree). Decoupled GNN varies only Σ; bundled GNNs co-vary $\psi/\Sigma/\varphi$. Sub-task assignments per regime are shown.

Table 2. Brute-force validation. DARTS hybrid consensus vs random ensembles.

Configuration	DARTS Hybrid		Random Ensemble		Fisher p	
	Val rank	Test rank	Val rank	Test rank	DARTS	Random
Peptides-struct SS1 (k=16)	4	4	—	—	—	—
Peptides-struct SS2 (k=9)	1	3	8	5	1.46×10^{-2}	0.81
Peptides-func SS1 (k=16)	2	**1**	—	—	—	—
Peptides-func SS2 (k=9)	1	2	2	1†	4.56×10^{-4}	0.81
MalNet-Tiny SS2 (k=81)	**29**	**26**	76	64	$\mathbf{4.09\times10^{-8}}$	0.62

† Random lands at test rank 1 by chance in this $k = 9$ space; Fisher $p = 0.81$ confirms no signal. SS1 random not run.

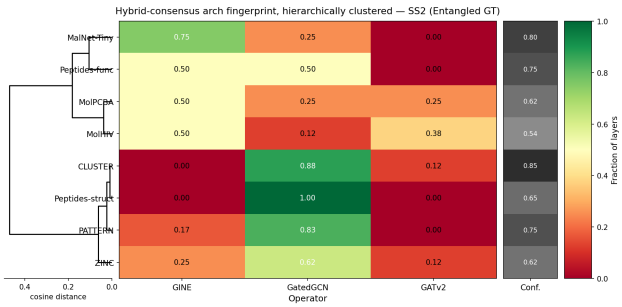


Figure 5. Cluster dendrogram SS2 + per-layer confidence column. Datasets group by per-layer fingerprint, not domain. SS1 in Figure 22.

Q3: Geometry probes aggregation behaviour. Real benchmarks confound geometry, task, and feature distribution. The synthetic experiments (Section 2.10) hold task and features fixed and vary only geometry; per-operator MSE, stability, spread, and winner for both settings are reported in Table 3. (1) *Decoupled*: the winning aggregator shifts with geometry, with no meaningful separation on the flat G_{clique} (Sum, Mean, and Degree-norm effectively tied), Sum lowest on G_{grid}, and Degree-norm clearly best on G_{tree}; the inter-operator spread Δ grows monotonically from clique to grid to tree. (2) *Bundled* (Appendix G): the same regime-dependence of operator preference holds, but the spread is non-monotonic in curvature, comparable on G_{clique} and G_{tree} and smallest on G_{grid}, unlike the monotonic growth of the decoupled setting. (3) *Per-node task dependence on the tree* (Appendix H, Figures 10 and 11): changing which node I predicts shifts operator preference at L, I, and R, while Mean continues to fail at R across all conditions. (4) *Root-node operator stress*: the tree root carries the deepest bottleneck, and the best-to-worst operator MSE gap there reaches $5\times$ in the decoupled setting and $14\times$ in the bundled setting, with Mean (decoupled) and GAT (bundled) failing sharply at R on G_{tree} (Figures 3 and 9). (5) *Per-node task dependence on the grid* (Appendix H and Figure 11): switching which node m predicts moves where MSE concentrates, since m tunes its representation toward whichever node it predicts. Thus it lowers the error of the node that shares that target while stress lands on the other. (6) Across all of the above, across seed variance is a sec-

Table 3. Synthetic-experiment results: per-operator mean MSE and across-seed Std (the stability axis) on the three regimes, decoupled and bundled ($N = 10$ seeds). Δ is the inter-operator spread (max−min mean MSE); the best operator per regime is in **bold**.

Operator	G_{clique} MSE	G_{clique} Std	G_{grid} MSE	G_{grid} Std	G_{tree} MSE	G_{tree} Std
Decoupled (only Σ varies)						
Sum	**0.856**†	0.061	**0.860**	0.092	0.274	0.329
Mean	**0.856**†	0.061	1.022	0.072	0.641	0.283
Degree-norm	**0.856**†	0.061	0.986	0.080	**0.261**	0.049
Max	0.920	0.061	0.903	0.061	0.372	0.079
Δ (spread)	0.064		0.162		0.380	
Winner	Tie†		Sum		Deg-norm	
Bundled (ψ, Σ, φ co-vary)						
GIN	0.497	0.102	**0.929**	0.074	**0.223**	0.064
GCN	0.856	0.061	1.052	0.072	0.369	0.074
GAT	**0.390**	0.033	1.079	0.088	0.696	0.165
SAGE-mean	0.476	0.056	1.037	0.087	0.284	0.053
SAGE-max	0.579	0.077	0.942	0.060	0.247	0.038
Δ (spread)	0.466		0.149		0.474	
Winner	GAT		GIN		GIN	

† Sum, Mean, and Degree-norm are statistically indistinguishable on G_{clique} (mean MSE 0.856 to three decimals); the regime is flat.

Table 4. SOTA performance with hybrid consensus on Entangled-GT (Part 1: ZINC, MalNet, PATTERN, CLUSTER). Best per dataset in **bold**.

Model	ZINC MAE↓	MalNet Acc↑	PATTERN Acc↑	CLUSTER Acc↑
Best GNN	0.079±.006	0.922±.0065	85.57±.09	73.84±.33
Best GT (GPS)	**0.070±.004**	**0.926±.0078**	86.69±.06	78.02±.18
Best GT-NAS	0.075±.07	—	**86.89±.02**	**79.24±.22**
Ours	**0.07±.004**	**0.926±.006**	86.65±.003	78.46±.015

Table 5. SOTA performance with hybrid consensus on Entangled-GT (Part 2: MolHIV, MolPCBA, Pep-Func, Pep-Struct). Best per dataset in **bold**.

Model	MolHIV AUROC↑	MolPCBA AP↑	Pep-Func AP↑	Pep-Struct MAE↓
Best GNN	**0.809±.006**	**0.299±.003**	0.593±.002	0.342±.001
Best GT (GPS)	0.788±.010	0.291±.003	0.654±.004	0.250±.001
Best GT-NAS	0.800±.0051	—	**0.667±.007**	**0.247±.002**
Ours	0.762±.01	0.261±.002	**0.662±.005**	**0.247±.001**

ond axis of operator differentiation alongside mean MSE, with operators that match in mean diverging substantially in stability. **Together, these point to operator preference as a multi-axis phenomenon: shaped by geometry, the considered operator pool, the task each node carries, and the architectural envelope, with stability acting as a second observable beyond mean performance.**

Hybrid consensus architectures match or approach SOTA. As a by-product, Tables 4 and 5 report our hybrid-consensus framework on Entangled-GT architecture against the best GT, GNN, and GT-NAS baselines per benchmark. We match SOTA on Peptides-struct and ZINC, approach SOTA on Peptides-func, CLUSTER, PATTERN, and MalNet, and underperform on MolHIV and MolPCBA. Full per-baseline tables in Appendix M. Per-layer heterogeneity details for all eight datasets are shown in Appendix V (Figures 20 and 21, Entangled-GT).

4. Conclusion

Implications for GFMs. Our results compose into a simple chain: operator preference varies with graph geometry, and graph geometry itself varies across hops, so the operator a layer wants varies with depth, as we observe across real-graph datasets, which points to it being geometry-structured. Because this per-layer signal is statistically validated rather than optimisation noise, it becomes a degree of freedom that current graph foundation models (GFMs) (Liu et al., 2024; He et al., 2025b; Wang et al., 2024; Xia & Huang, 2024; Galkin et al., 2024) leave unused. The fixed-backbone assumption occupies a measure-vanishing corner of the architecture space (Proposition 2.3) and is sensitive to the wrapping envelope (Remark 2.6). Adaptive GFMs already vary *which* operator they use but repeat it across depth, so per-layer preference is a complementary axis alongside per-input (AutoGFM (Chen et al., 2025)), per-node (GraphAny (Zhao et al., 2025)), and per-task by range (GOBLIN (Gutteridge et al., 2026)), distinguished by being conditioned on geometry. One route to it is learned, AutoGFM-style routing, but the per-seed NAS signal is noisy and needs many seeds to converge, which is costly at scale. A more durable target is a principled mapping, even an approximate or range-valued one, from a geometric descriptor to operator preference for a given operator set. Our per-node results suggest that this mapping is finer than the layer itself: preference shifts with the bottleneck position and with the roles of the surrounding and connecting nodes. A geometry-aware backbone could then apply geometry-determined operators per node and per hop without retraining, with learnable components still in play but grounded in a principled basis.

Acknowledgements

This research was funded by Google DeepMind, with S.C. being supported by the Google DeepMind Scholarship for her DPhil studies. The authors acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work.https://doi.org/10.5281/zenodo.22558.

References

Baek, J., Kang, M., and Hwang, S. J. Accurate learning of graph representations with graph multiset pooling. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Bamberger, J., Barbero, F., Dong, X., and Bronstein, M. M. Bundle neural network for message diffusion on graphs. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a.

Bamberger, J., Gutteridge, B., le Roux, S., Bronstein, M. M., and Dong, X. On measuring long-range interactions in graph neural networks. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b.

Barbero, F., Velingker, A., Saberi, A., Bronstein, M. M., and Giovanni, F. D. Locality-aware graph rewiring in gnns. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Beaini, D., Passaro, S., Létourneau, V., Hamilton, W. L., Corso, G., and Lió, P. Directional graph networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 748–758. PMLR, 2021.

Bodnar, C., Frasca, F., Otter, N., Wang, Y., Liò, P., Montúfar, G. F., and Bronstein, M. M. Weisfeiler and lehman go cellular: CW networks. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 2625–2640, 2021.

Bouritsas, G., Frasca, F., Zafeiriou, S., and Bronstein, M. M. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):657–668, 2023.

Bresson, X. and Laurent, T. Residual gated graph convnets. *CoRR*, abs/1711.07553, 2017.

Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Chakraborty, S. S. and Minary, P. DARTS-GT: differentiable architecture search for graph transformers with quantifiable instance-specific interpretability analysis. *CoRR*, abs/2510.14336, 2025.

Chen, D., O'Bray, L., and Borgwardt, K. M. Structure-aware transformer for graph representation learning. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3469–3489. PMLR, 2022.

Chen, H., Wang, X., Zhang, Z., Li, H., Feng, L., and Zhu, W. Autogfm: Automated graph foundation model with adaptive architecture customization. In Singh, A., Fazel, M., Hsu, D., Lacoste-Julien, S., Berkenkamp, F., Maharaj, T., Wagstaff, K., and Zhu, J. (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Velickovic, P. Principal neighbourhood aggregation for graph nets. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *AAAI Workshop on Deep Learning on Graphs: Methods and Applications*, 2021.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a.

Dwivedi, V. P., Rampášek, L., Galkin, M., Parviz, A., Wolf, G., Luu, A. T., and Beaini, D. Long range graph benchmark. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022b.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Finkelshtein, B., Ceylan, I. I., Bronstein, M. M., and Levie, R. Equivariance everywhere all at once: A recipe for graph foundation models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Galkin, M., Yuan, X., Mostafa, H., Tang, J., and Zhu, Z. Towards foundation models for knowledge graph reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Gao, Y., Yang, H., Zhang, P., Zhou, C., and Hu, Y. Graphnas: Graph neural architecture search with reinforcement learning. *CoRR*, abs/1904.09981, 2019.

Giovanni, F. D., Giusti, L., Barbero, F., Luise, G., Lio, P., and Bronstein, M. M. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7865–7885. PMLR, 2023.

Gutteridge, B., Dong, X., Bronstein, M. M., and Giovanni, F. D. Drew: Dynamically rewired message passing with delay. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12252–12267. PMLR, 2023.

Gutteridge, B., Bronstein, M. M., and Dong, X. Can graph foundation models generalize over architecture? In *ICLR 2026 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2026.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034, 2017.

He, C., Chen, J., Li, Q., Wang, Y., and Gao, J. Decoupled graph neural architecture search with explainable variable propagation operation. *Knowl. Inf. Syst.*, 67(5):4677–4702, 2025a.

He, Y., Sui, Y., He, X., and Hooi, B. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In Sun, Y., Chierichetti, F., Lauw, H. W., Perlich, C., Tok, W. H., and Tomkins, A. (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 448–459. ACM, 2025b.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V. S., and Leskovec, J. Strategies for pre-training graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Hussain, M. S., Zaki, M. J., and Subramanian, D. Global self-attention as a replacement for graph convolution. In Zhang, A. and Rangwala, H. (eds.), *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 655–665. ACM, 2022.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Kreuzer, D., Beaini, D., Hamilton, W. L., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 21618–21629, 2021.

Lai, K., Zha, D., Zhou, K., and Hu, X. Policy-gnn: Aggregation optimization for graph neural networks. In Gupta, R., Liu, Y., Tang, J., and Prakash, B. A. (eds.), *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pp. 461–471. ACM, 2020.

Lin, J., Guo, X., Zhang, S., Zhou, D., Zhu, Y., and Shun, J. Unifiedgt: Towards a universal framework of transformers in large-scale graph learning. In Ding, W., Lu, C., Wang, F., Di, L., Wu, K., Huan, J., Nambiar, R., Li, J., Ilievski, F., Baeza-Yates, R., and Hu, X. (eds.), *IEEE International Conference on Big Data, BigData 2024, Washington, DC, USA, December 15-18, 2024*, pp. 1057–1066. IEEE, 2024.

Lin, K., Wang, L., and Liu, Z. Mesh graphormer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 12919–12928. IEEE, 2021.

Liu, H., Simonyan, K., and Yang, Y. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Liu, H., Feng, J., Kong, L., Liang, N., Tao, D., Chen, Y., and Zhang, M. One for all: Towards training one graph model for all classification tasks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Liu, Y. and Liu, J. A surrogate evolutionary neural architecture search algorithm for graph neural networks. *Appl. Soft Comput.*, 144:110485, 2023.

Qin, Y., Wang, X., Zhang, Z., and Zhu, W. Graph differentiable architecture search with structure learning. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 16860–16872, 2021.

Qin, Y., Wang, X., Zhang, Z., Xie, P., and Zhu, W. Graph neural architecture search under distribution shifts. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18083–18095. PMLR, 2022a.

Qin, Y., Zhang, Z., Wang, X., Zhang, Z., and Zhu, W. Nas-bench-graph: Benchmarking graph neural architecture search. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022b.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515. Curran Associates, Inc., 2022.

Sasaki, Y. Explainable graph neural architecture search via monte-carlo tree search. In *IEEE International Conference on Big Data, BigData 2025, Macau, China, December 8-11, 2025*, pp. 752–757. IEEE, 2025.

Shi, M., Tang, Y., Zhu, X., Huang, Y., Wilson, D. A., Zhuang, Y., and Liu, J. Genetic-gnn: Evolutionary architecture search for graph neural networks. *Knowl. Based Syst.*, 247:108752, 2022.

Song, W., Wu, X., Yang, B., Zhou, Y., Xiao, Y., Liang, Y., Ge, H., Lee, H. P., and Wu, C. Towards efficient few-shot graph neural architecture search via partitioning gradient contribution. In Antonie, L., Pei, J., Yu, X., Chierichetti, F., Lauw, H. W., Sun, Y., and Parthasarathy, S. (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pp. 2609–2620. ACM, 2025.

Tönshoff, J., Ritzert, M., Wolf, H., and Grohe, M. Walking out of the weisfeiler leman hierarchy: Graph learning beyond message passing. *Trans. Mach. Learn. Res.*, 2023, 2023.

Tönshoff, J., Ritzert, M., Rosenbluth, E., and Grohe, M. Where did the gap go? reassessing the long-range graph benchmark. *Trans. Mach. Learn. Res.*, 2024, 2024.

Topping, J., Giovanni, F. D., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Wang, C., Zhao, J., Li, L., Jiao, L., Liu, F., and Yang, S. Automatic graph topology-aware transformer. *IEEE Trans. Neural Networks Learn. Syst.*, 36(5):8470–8484, 2025a.

Wang, H., Yin, H., Zhang, M., and Li, P. Equivariant and stable positional encoding for more powerful graph neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Wang, H., Gao, Y., Zheng, X., Zhang, P., Bu, J., and Yu, P. S. Graph neural architecture search with large language models. *Sci. China Inf. Sci.*, 68(12), 2025b.

Wang, Q., Shao, Y., Ma, Y., and Liu, M. Nodenas: Node-specific graph neural architecture search for out-of-distribution generalization. In Zhu, F., Yu, P. S., Nadamoto, A., Lim, E., Shim, K., Ding, W., and Zhang, B. (eds.), *Database Systems for Advanced Applications*

- *30th International Conference, DASFAA 2025, Singapore, Singapore, May 26-29, 2025, Proceedings, Part III*, volume 15988 of *Lecture Notes in Computer Science*, pp. 87–102. Springer, 2025c.

Wang, Z., Zhang, Z., Chawla, N. V., Zhang, C., and Ye, Y. GFT: graph foundation model with transferable tree vocabulary. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Wei, L., Zhao, H., Yao, Q., and He, Z. Pooling architecture search for graph classification. In Demartini, G., Zuccon, G., Culpepper, J. S., Huang, Z., and Tong, H. (eds.), *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pp. 2091–2100. ACM, 2021.

White, C., Nolen, S., and Savani, Y. Exploring the loss landscape in neural architecture search. In de Campos, C. P., Maathuis, M. H., and Quaeghebeur, E. (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pp. 654–664. AUAI Press, 2021.

Wu, Z., Jain, P., Wright, M. A., Mirhoseini, A., Gonzalez, J. E., and Stoica, I. Representing long-range context for graph neural networks with global attention. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 13266–13279, 2021.

Xia, L. and Huang, C. Anygraph: Graph foundation model in the wild. *CoRR*, abs/2408.10700, 2024.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yang, M., Shen, Y., Qi, H., and Yin, B. Breaking the expressive bottlenecks of graph neural networks. *CoRR*, abs/2012.07219, 2020.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T. Do transformers really perform badly for graph representation? In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28877–28888, 2021.

Zela, A., Elsken, T., Saikia, T., Marrakchi, Y., Brox, T., and Hutter, F. Understanding and robustifying differentiable architecture search. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Zhang, Z., Wang, X., Guan, C., Zhang, Z., Li, H., and Zhu, W. Autogt: Automated graph transformer architecture search. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Zhao, H., Yao, Q., and Tu, W. Search to aggregate neighborhood for graph neural network. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pp. 552–563. IEEE, 2021.

Zhao, J., Zhu, Z., Galkin, M., Mostafa, H., Bronstein, M. M., and Tang, J. Fully-inductive node classification on arbitrary graphs. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Zhao, L., Jin, W., Akoglu, L., and Shah, N. From stars to subgraphs: Uplifting any GNN with local structure awareness. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Impact Statement

This paper studies whether layer-wise architectural heterogeneity in graph neural networks is a statistically real and geometrically responsive phenomenon, using neural architecture search as a diagnostic instrument. The contribution is methodological and analytical rather than the deployment of a new system. The experiments use established public graph benchmarks and controlled synthetic graphs, involve no human subjects or personal data, and raise no direct privacy or fairness concerns specific to this work.

A. Use of LLMs and Other Tools

We used GPT-5, Gemini, and Claude to check for mathematical and notational inconsistencies and to assist with code. We use Grammarly to refine writing.

B. Related Work

Graph Transformers. A broad line of recent work combines global self-attention with graph inductive bias (Wu et al., 2021; Wang et al., 2025a; Chakraborty & Minary, 2025). GraphGPS (Rampášek et al., 2022) interleaves local message passing with global attention and aggregates rich positional/structural encodings, effectively running a parallel MPNN-attention design and setting a strong template for hybrid GTs. SAN (Kreuzer et al., 2021) injects spectral positional information and runs a fully-connected Transformer, showing that learned spectral PEs can offset over-squashing while preserving structure. SAT (Chen et al., 2022) makes attention itself structure-aware by extracting k-subtree/k-subgraph features with a base GNN before computing attention, consistently improving over baseline GTs. Graphormer (Ying et al., 2021) augments attention with centrality, shortest-path, and edge encodings, achieving state-of-the-art molecular property prediction at scale. Variants in other modalities (e.g. Mesh Graphormer (Lin et al., 2021)) reinforce the pattern of mixing graph convolution with attention to capture local-and-global interactions. Graph-Trans (Wu et al., 2021) integrates a preliminary stack of GNN layers before Transformer blocks, using the GNN to encode local structural patterns that the subsequent attention layers can exploit.

Within hybrid GT designs, we use a per-layer asymmetric formulation in which the GNN supplies keys and values to attention, while queries are derived from node features. Our work adopts this backbone as one of three architectural envelopes (Section 2.5), but the central question is not the attention design — it is whether the per-layer GNN choice itself is a structurally meaningful variable, examined here through the lens of layer-wise heterogeneity rather than as an attention or pooling improvement.

Geometry, over-squashing, and rewiring. Topping et al. (2022) formalised graph-structural bottlenecks via Balanced Forman Curvature (BFC), an edge-local descriptor that quantifies how negative curvature concentrates along bottleneck edges. Giovanni et al. (2023) extended this analysis to show that graph topology — in particular, nodes at high commute time drive over-squashing in message-passing networks, with sensitivity bounds that depend on width, depth, and topology jointly. A line of follow-up work addresses over-squashing through rewiring: Barbero et al. (2024) proposes locality-aware rewiring to preserve structure while alleviating bottlenecks; Gutteridge et al. (2023) introduces dynamically rewired message passing with delay (DRew) that conditions message routing on geodesic distance; Bamberger et al. (2025a) proposes Bundle Neural Networks (BuNN) for message diffusion on graphs, removing the need for explicit rewiring through flat-reactivity (geometrically robust) primitives. Range as a measurement primitive for long-range interactions in GNNs is studied in Bamberger et al. (2025b). Long-range benchmarks (Dwivedi et al., 2022b), with subsequent reassessments (Tönshoff et al., 2024), anchor the empirical study of bottleneck phenomena. Our work builds on this framework by reframing the question from *whether* message-passing capacity is preserved (their question) to *which* aggregation operator is preferred in each geometric regime, with BFC as the descriptor we condition on (which can be extended to other reliable geometric descriptors as well).

NAS and DARTS. Differentiable architecture search (DARTS) (Liu et al., 2019) relaxes discrete architecture choice into a continuous mixed-operator parameterised by architecture parameters, enabling gradient-based search over the space. Subsequent work documented serious instabilities: DARTS is sensitive to initialisation and can collapse to skip-dominated solutions or produce inconsistent architectures across seeds (Zela et al., 2020). The single-seed reporting practice common in NAS conceals this variability, and remedies have included perturbation-based search, multi-seed ensembling, and operator-wise stability analyses. Our ensemble-consensus protocol extends the multi-seed line with explicit per-layer confidence

margins, a hybrid voting-and-mean-fallback rule, and complementary statistical tests to certify that the per-layer signal is real.

Graph NAS. Early Graph NAS (GNAS) studies search over message-passing cells on pure-GNN backbones: SANE (Zhao et al., 2021) searches per-layer aggregation choices via differentiable methods, GraphNAS (Gao et al., 2019) uses RL-based search for GNN architectures, and GASSO (Qin et al., 2021) couples GNN search with structural learning. Pooling-architecture search PAS (Wei et al., 2021) extends NAS to graph-pooling operations. Decoupled GNAS (He et al., 2025a) separates the search of feature propagation from feature transformation, enabling explainable variable propagation. Structure-learning and graph-DARTS variants (Qin et al., 2021) combine architecture search with simultaneous graph-structure refinement. Explainable Monte-Carlo tree-search GNAS (Sasaki, 2025) provides interpretability over the search trajectory. NodeNAS (Wang et al., 2025c) introduces node-specific architecture search for OOD generalisation. (Wang et al., 2025b) uses LLMs to assist the search instead while (Shi et al., 2022) uses a genetic evolutionary algorithm to perform the same. Our contribution differs from this line in two ways: (i) we use NAS as a *diagnostic instrument* to study whether the per-layer signal is structural, rather than as a method to optimise a single configuration; and (ii) we operate within a Graph Transformer envelope, where the per-layer GNN is one component of a larger architecture rather than a stand-alone backbone.

NAS for Graph Transformers. AutoGT (Zhang et al., 2023) defines a unified GT formulation and searches Transformer hyperparameters plus structural/positional encodings, but the resulting architecture is globally tied rather than layer-wise heterogeneous. EGTAS (Wang et al., 2025a) expands the GT search space with macro- (topology) and micro- (graph-aware strategies) design and uses evolutionary search to assemble end-to-end GTs. UGAS (Song et al., 2025) proposes a Unified Graph NAS exploring which subsets of MPNN and GT modules to activate in parallel per layer, with GTs treated as optional rather than foundational. UNIFIEDGT (Lin et al., 2024) modularises GT components (sampling, structural priors, attention, local/global mixing, FFNs) to instantiate many GT variants. Our work is orthogonal: we keep the GT spine and search for the local GNN operator for each layer using ensemble differentiable NAS, yielding adaptive, heterogeneous architectures while preserving global attention. This isolates the per-layer GNN choice as an independent variable, which is a precondition for studying whether its variation is structurally meaningful.

Graph foundation models and architecture adaptivity. The dominant graph foundation model lineage (Liu et al., 2024; He et al., 2025b; Wang et al., 2024; Xia & Huang, 2024; Galkin et al., 2024) relies on fixed-backbone architectures, typically a GraphGPS-style or pure-attention spine repeated identically across layers and datasets. Recent work questions this along three axes of architectural adaptivity: per-input (AutoGFM (Chen et al., 2025) customises architecture conditional on the input graph and produces per-layer heterogeneity as a by-product), per-node (GraphAny (Zhao et al., 2025) mixes a fixed set of linear graph operators with node-specific weights), and per-task (GOBLIN (Gutteridge et al., 2026) adapts by conditioning on geodesic and resistance distance). We add a complementary axis: per-layer operator preference as a geometry-grounded phenomenon. The granularity at which architectures should adapt, and the structural property on which adaptation should be conditioned, remain open questions, and the present paper addresses the precondition — whether the per-layer signal is real, robust, and shows directional dependence on graph geometry.

C. Limitations

Geometry-to-operator mapping is empirical, not formal. The geometry-to-operator preference relationship (Q3) is established empirically through controlled synthetic experiments and is not framed as a formal theoretical claim. We do not derive a functional form for the mapping ϕ from bottleneck pressure to preferred operator, nor do we prove that such a mapping exists deterministically across all envelopes, descriptors, and operator pools. A formal proof would require either deriving operator-output sensitivity to source signal as a function of curvature (extending the Jacobian sensitivity bounds of Topping et al. (2022) and Giovanni et al. (2023) to operator selection) or an explicit characterisation of ϕ. The paper provides directional evidence and makes the question tractable, but does not resolve it.

Synthetic generalisation. The synthetic experiments use three extreme geometric regimes (clique, grid, binary tree) chosen to span the BFC spectrum from $+1.11$ to -0.33. These are minimal-graph constructions, not representative of the full diversity of real graph topologies. The bundled experiments also show non-monotonic inter-operator spread across topologies, suggesting that the relationship between BFC and operator preference is more complex than a clean curvature-to-spread monotonicity, particularly when the message function ψ and update φ co-vary with the aggregation operator Σ rather than being held fixed. A larger and more diverse synthetic suite, with finer-grained curvature gradients and

intermediate regimes, would strengthen the generalisation argument.

NAS and search constraints. The ensemble-consensus protocol relies on DARTS as the underlying search algorithm, which has known instabilities: DARTS is sensitive to initialisation and can collapse to skip-dominated solutions in some regimes (Zela et al., 2020). The hybrid voting and mean-fallback rule mitigates this but does not eliminate it. The main experiments use $N = 10$ seeds. We additionally assess robustness with $N = 20$ on MalNet-Tiny, where the consensus architecture remains unchanged, and with higher-N analyses on Peptides ($N = 38$ per task; pooled $N = 76$ across both tasks). Nevertheless, weak per-layer preferences may remain undetected at $N = 10$. The operator pools we consider are also modest (SS1: 4 operators, SS2: 3 operators); larger pools, including operators outside the message-passing/transformer paradigms (e.g. diffusion-based, spectral, sheaf-theoretic), might reveal different per-layer patterns that our search space cannot express.

Compute cost. The ensemble protocol runs $N = 10$ DARTS searches per (dataset, search space, backbone) configuration; across 31 configurations on eight benchmarks, the cumulative compute is substantial. While each individual DARTS run is comparable to a single training run, the ensemble multiplier limits the practicality of applying this protocol to very large graph datasets. The brute-force validation we report on five small-enough configurations is a sanity check rather than a routine validation step — on the full eight benchmarks, brute-force is computationally infeasible.

Granularity. The paper operates at per-layer granularity for operator selection. Per-node breakdowns on the binary tree show that operator stress concentrates at specific bottleneck positions (root vs. leaves), suggesting that finer-grained per-node operator selection might capture structure that per-layer selection misses. Whether per-node selection is the right granularity and whether it would be tractable on real graphs remain open questions. The paper establishes per-layer heterogeneity as a phenomenon but does not investigate finer granularities empirically.

D. Proof and Discussion of Proposition 2.3

We restate Proposition 2.3 and provide its proof, numerical illustrations, and the empirical anchor in our experiments.

Proposition D.1 (Measure of Homogeneity, restated)**.** *Let $\mathcal{O} = \{o_1, \ldots, o_n\}$ denote the operator pool with $n = |\mathcal{O}|$, and let L denote the number of layers. The architecture search space $\mathcal{S} = \mathcal{O}^L$ has the homogeneous subset $\mathcal{H}_{\text{homo}} = \{(o, o, \ldots, o) : o \in \mathcal{O}\}$, occupying a vanishing fraction:*

$$\frac{|\mathcal{H}_{\text{homo}}|}{|\mathcal{S}|} = \frac{n}{n^L} = n^{1-L}. \tag{8}$$

Proof. The homogeneous subset is parameterised by a single operator choice $o \in \mathcal{O}$ shared across all L layers, giving $|\mathcal{H}_{\text{homo}}| = n$. The full space has $|\mathcal{S}| = n^L$ due to independent choices at each layer. The ratio follows directly. □

Numerical illustration. For $n = 4$ operators and $L = 6$ layers, the homogeneous fraction is $4^{-5} = 1/1024 \approx 0.001$. For $n = 3$ operators and $L = 6$ layers, the fraction is $3^{-5} = 1/243 \approx 0.004$. For $n = 2$ operators and $L = 3$ layers, the fraction is $2^{-2} = 0.25$. The fraction shrinks exponentially as L grows: at depth $L = 10$ with $n = 4$, the homogeneous fraction is approximately 4×10^{-6}.

Empirical anchor. This counting result matches our observation on MalNet-Tiny. The top-performing architectures across the discrete landscape are overwhelmingly heterogeneous and grow with the number of layers l (Figure 6). Moreover, it has been observed that the top-performing homogeneous architecture exhibits substantial variability in stability across test and val set performance compared to our consensus architecture (Figure 18) in Malnet-Tiny.

E. Proof and Discussion of Proposition 2.5

We restate Proposition 2.5 and provide its full proof, numerical illustrations, three remarks on robustness, and the empirical anchor.

Proposition E.1 (Local Minima Distribution, restated)**.** *Equip $\mathcal{S} = \mathcal{O}^L$ with the one-layer-flip neighbourhood structure, and let $\mathcal{L} : \mathcal{S} \to \mathbb{R}$ denote validation loss. Under Assumption 2.4, the expected number of heterogeneous local minima*

exceeds the expected number of homogeneous local minima by a factor that grows exponentially in L:

$$\frac{\mathbb{E}\big[\,\#\{\textit{heterogeneous local minima}\}\,\big]}{\mathbb{E}\big[\,\#\{\textit{homogeneous local minima}\}\,\big]} \;=\; n^{L-1} - 1. \tag{9}$$

Proof. Fix an architecture $\pi \in \mathcal{S}$. Under Assumption 2.4, the loss values on the closed neighbourhood $\overline{\mathcal{N}}(\pi)$ — which has size $L(n-1)+1$ are exchangeable. By symmetry, each architecture in $\overline{\mathcal{N}}(\pi)$ is equally likely to attain the minimum value within this set, so

$$\mathbb{P}\big[\,\pi \text{ is a local minimum}\,\big] \;=\; \frac{1}{L(n-1)+1}. \tag{10}$$

This per-architecture probability is identical across all $\pi \in \mathcal{S}$ because every architecture has a closed neighbourhood of the same size. By linearity of expectation, for any subset $X \subseteq \mathcal{S}$,

$$\mathbb{E}\big[\,\#\{\text{local minima in } X\}\,\big] \;=\; \sum_{\pi \in X} \mathbb{P}[\,\pi \text{ is a local minimum}\,] \;=\; \frac{|X|}{L(n-1)+1}. \tag{11}$$

Applying this to $\mathcal{H}_{\text{homo}}$ (with $|\mathcal{H}_{\text{homo}}| = n$) and $\mathcal{H}_{\text{het}}$ (with $|\mathcal{H}_{\text{het}}| = n^L - n$):

$$\mathbb{E}\big[\,\#\{\text{homogeneous local minima}\}\,\big] = \frac{n}{L(n-1)+1}, \tag{12}$$

$$\mathbb{E}\big[\,\#\{\text{heterogeneous local minima}\}\,\big] = \frac{n^L - n}{L(n-1)+1}. \tag{13}$$

Taking the ratio cancels $L(n-1)+1$ and gives $(n^L - n)/n = n^{L-1} - 1$. □

Numerical illustration. For $n = 4$ operators and $L = 6$ layers, the per-architecture probability of being a local minimum is $1/19 \approx 0.053$, the expected total number of local minima is $4^6/19 \approx 215$, of which approximately $4/19 \approx 0.2$ are homogeneous and approximately 215 are heterogeneous — a ratio of $4^5 - 1 = 1023$. For $n = 3$ and $L = 4$, the ratio is $3^3 - 1 = 26$. For $n = 2$ and $L = 3$ (the smallest non-trivial case), heterogeneous local minima still outnumber homogeneous ones by a factor of $2^2 - 1 = 3$.

Remark E.1 (Flat landscapes). If the loss surface contains flat regions where many architectures share near-identical loss, Assumption 2.4 breaks down within those regions. However, Proposition 2.3 (counting) continues to apply: provided flat regions are not systematically biased toward the homogeneous diagonal, heterogeneous architectures occupy an overwhelmingly larger proportion of any such region in $\mathcal{S}$, so a search procedure landing in it still lands heterogeneous with high probability.

Remark E.2 (Non-convex landscapes). Real loss surfaces on $\mathcal{O}^L$ are highly non-convex, with many local minima. Proposition 2.5's conclusion is amplified rather than weakened in this regime: each additional local minimum is a draw from a dominantly heterogeneous architecture space, so more local minima yield more heterogeneous outcomes for search procedures.

Remark E.3 (Correlated losses). Real loss surfaces also have correlation structure: architectures differing in one operator typically have similar losses. The exchangeability assumption is therefore a simplification. The continuous-PDF framework of White et al. (2021) handles correlated losses via a local PDF $\mathrm{pdf}_e(x, y)$ between neighbouring loss values. Under correlated losses, the absolute number of local minima decreases, but the homogeneous-vs-heterogeneous *ratio* survives qualitatively because $|\mathcal{H}_{\text{het}}|/|\mathcal{H}_{\text{homo}}| = n^{L-1} - 1$ is determined by the architecture space's combinatorial structure, not by the loss correlation. Hence, the qualitative claim that local minima are predominantly heterogeneous — is robust.

F. Discussion of Remark 2.6

We discuss Remark 2.6 in more detail, including the informal reasoning, an overlap heuristic, the consequence for consensus architectures, and the SS1 robustness figure.

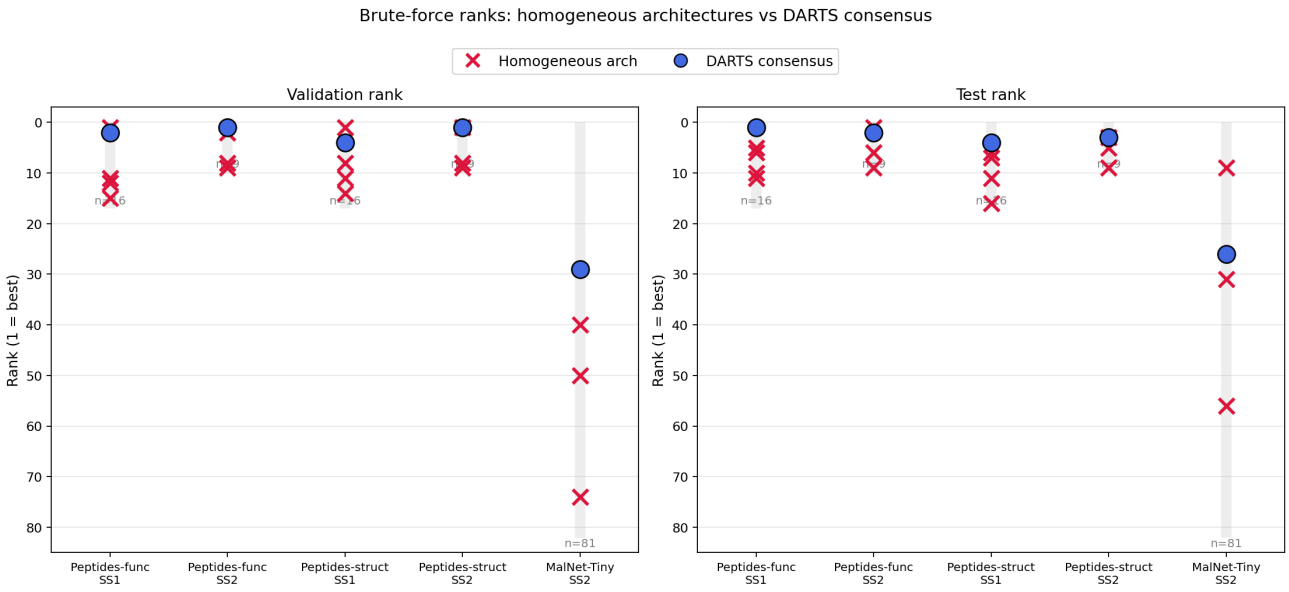


Figure 6. Empirical anchor: Brute-force ranks of homogeneous architectures (×) and DARTS consensus architectures (•) across five (dataset, search space) configurations. For each configuration, all $|\mathcal{O}|^L$ architectures were trained from scratch (3 seeds each) and ranked by validation and test performance (rank 1 = best). DARTS consensus architectures place in the top tier on Peptides-func (val ranks 2/16 SS1, 1/9 SS2) and Peptides-struct (val ranks 4/16 SS1, 1/9 SS2), and at val rank 29/81 on MalNet-Tiny ($L = 4$). Homogeneous architectures occasionally appear in the top tier on shallow $L = 2$ configurations, consistent with Proposition 2.5, where the predicted heterogeneous-to-homogeneous local-minima ratio at $L = 2, n = 4$ is only $n^{L-1} - 1 = 3$; the dominance of heterogeneous architectures in the top tier becomes pronounced only as L grows.

Informal reasoning. The argument runs as follows. The loss $\mathcal{L}_B(\pi)$ is determined by the forward computation of the model with wrapper B on data, which is a function of the layer-input distributions $\{\mathcal{I}_\ell^B\}_{\ell=1}^L$. If $d_{\text{env}}(B_A, B_B) > 0$, there exists at least one layer $\ell^\star$ such that $\mathcal{I}_{\ell^\star}^{B_A} \neq \mathcal{I}_{\ell^\star}^{B_B}$ in distribution; the per-layer GNN at $\ell^\star$ therefore receives input from a different distribution under B_A versus B_B. Different input distributions to the same operator induce different output distributions and hence different gradients, so the relative ordering of $\mathcal{L}_{B_A}(\pi)$ across architectures π that differ only in their layer-$\ell^\star$ operator is in general different from the corresponding ordering under $\mathcal{L}_{B_B}$. This implies that the local-minimum status of architectures within those neighbourhoods generally differs between wrappers, so identical local-minima sets across B_A, B_B would require the layer-input distributions to match at every layer, contradicting $d_{\text{env}}(B_A, B_B) > 0$. A fully rigorous version would specify a probability model on loss surfaces conditional on the envelope and prove a quantitative gap. We defer this to future work, treating the present statement as a structural observation supported empirically by Figures 2 and 7.

Overlap heuristic. Combining Proposition 2.5 and Remark 2.6 suggests a heuristic on the overlap of local-minima sets. Under Assumption 2.4, the per-architecture probability of being a local minimum under any single wrapper is $1/(L(n-1)+1)$ (Proposition 2.5). When $d_{\text{env}}(B_A, B_B) > 0$, the two local-minimum events are not perfectly correlated — otherwise the two loss orderings would coincide which heuristically gives

$$\mathbb{P}[\,\pi \in \mathrm{LM}(B_A) \cap \mathrm{LM}(B_B)\,] \;<\; \mathbb{P}[\,\pi \in \mathrm{LM}(B_A)\,] \;=\; \frac{1}{L(n-1)+1}. \tag{14}$$

By linearity of expectation, $\mathbb{E}[|\mathrm{LM}(B_A) \cap \mathrm{LM}(B_B)|] < \mathbb{E}[|\mathrm{LM}(B_A)|]$. Quantifying how the gap scales with d_{env} would require additional regularity assumptions on how loss orderings depend on input distributions, and is left to future work.

Consequence for consensus architectures. A search procedure run inside envelope B_A converges to some $\pi_A^{\text{cons}} \in \mathrm{LM}(B_A)$, and similarly for B_B. Since $\mathrm{LM}(B_A) \cap \mathrm{LM}(B_B)$ is generically a strict subset of $\mathrm{LM}(B_A)$,

$$\mathbb{P}[\pi_A^{\text{cons}} \neq \pi_B^{\text{cons}}] \;>\; 0 \quad \text{when } d_{\text{env}}(B_A, B_B) > 0, \tag{15}$$

and in Hamming-distance terms, $\mathbb{E}[d_H(\pi_A^{\text{cons}}, \pi_B^{\text{cons}})] > 0$.

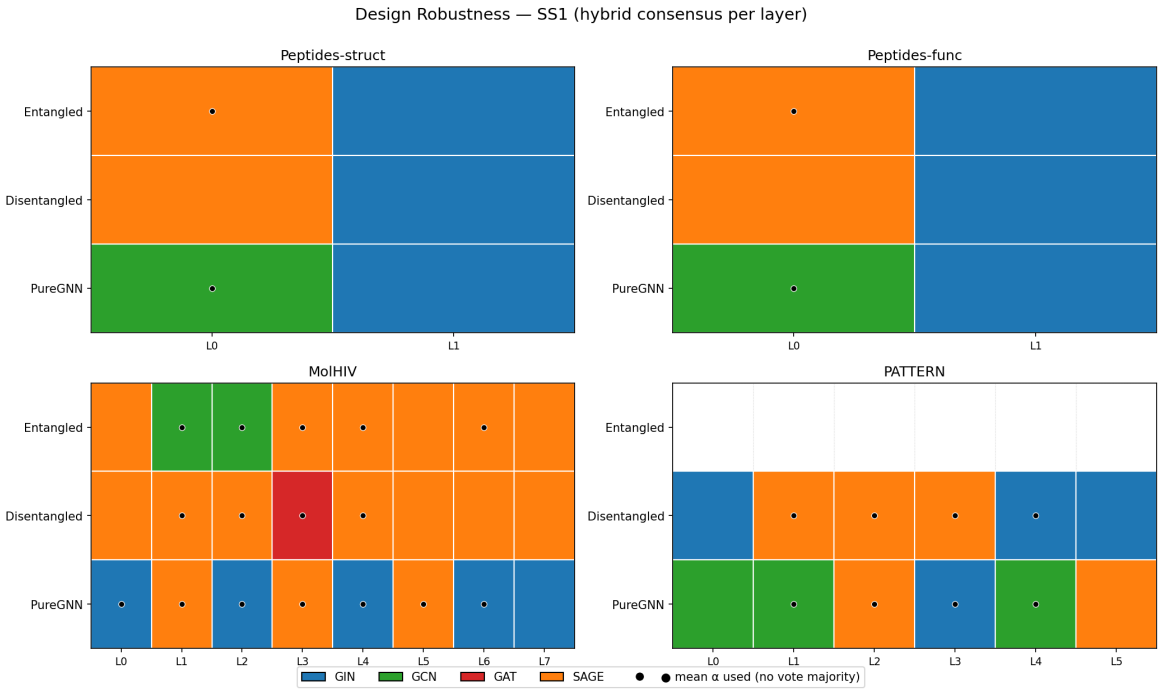


Figure 7. Per-layer consensus operator across three envelopes (Entangled-GT, DisEntangled-GT, Pure GNN) on datasets in SS1. The envelopes disagree on consensus operators at multiple layers within the same (dataset, search space) configuration, mirroring the SS2 pattern in Figure 2 and Remark 2.6

What this shows. Distinct wrappers induce distinct loss surfaces on the same architecture search space, with generically distinct local-minima sets, and consequently distinct consensus architectures. This explains why our three architectural envelopes (Entangled-GT, DisEntangled-GT, Pure GNN) — wrapping the same operator pool over the same data produces different consensus architectures at the same layer (Figure 2, Figure 7); and why the decoupled and bundled synthetic experiments — corresponding to two different abstraction levels of *envelope* applied to the same data produce different operator preferences at controlled geometric points (Section 2.10). Both findings are consistent with Remark 2.6.

G. Synthetic Experiments: Bundled-Setting Results

This appendix presents the bundled-setting results referenced in Section 2.10, where the entire GNN (ψ, Σ, φ) varies across the operator pool $\mathcal{A}_{\text{bun}} = \{\text{GIN}, \text{GCN}, \text{GAT}, \text{SAGE-mean}, \text{SAGE-max}\}$. Notation follows Section 2.10: $\mu_k(G) = \mathbb{E}[\text{MSE}_k(G)]$, inter-operator spread $\Delta(G) = \max_{i,j} |\mu_i(G) - \mu_j(G)|$, and best operator $k^\star(G) = \arg\min_k \mu_k(G)$.

Bundled overall results. The bundled experiments corroborate the decoupled-setting finding that operator preferences differ between geometric regimes: $k^\star(G_1) \neq k^\star(G_2)$ for distinct regimes. The inter-operator spread is non-monotonic across topologies, with $\Delta(G_{\text{clique}}) \approx 0.47$ and $\Delta(G_{\text{tree}}) \approx 0.47$, while $\Delta(G_{\text{grid}}) \approx 0.15$ is the smallest of the three, so the spread does not increase monotonically with curvature, unlike the decoupled setting.

Per-node operator stress on the binary tree. On G_{tree}, nodes occupy distinct geometric positions: the root sits at the deepest cumulative bottleneck (all leaf signals must converge through internal nodes to the root), internal nodes at intermediate positions, and leaves at the minimal cumulative bottleneck. Per-node MSE distributions across operators show up to $14\times$ MSE differences between best and worst operators at the root in the bundled setting (Figure 9), while leaves and internal nodes show milder differentiation.

G.1. Training Setup

Each configuration is trained for 500 epochs with Adam (lr = 0.001), a hidden dimension of 128, MSE loss, and 10 random seeds. Layer count is fixed at the minimum number of message-passing steps required for full geometric reach (1 for Clique,

2 for Grid, and 3 for Tree). All other components are held fixed across the 4 (decoupled) or 5 (bundled) variants within each regime, thereby isolating aggregation-induced (or operator-induced) effects.

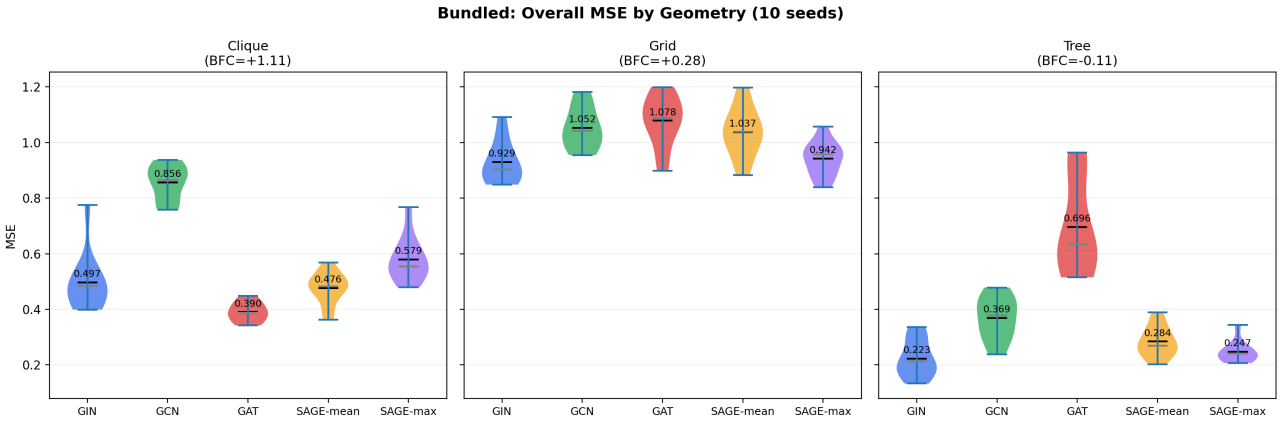


Figure 8. Bundled-setting MSE per operator across three graph topologies. Operator preferences differ between regimes; $\Delta(G)$ is non-monotonic across topologies.

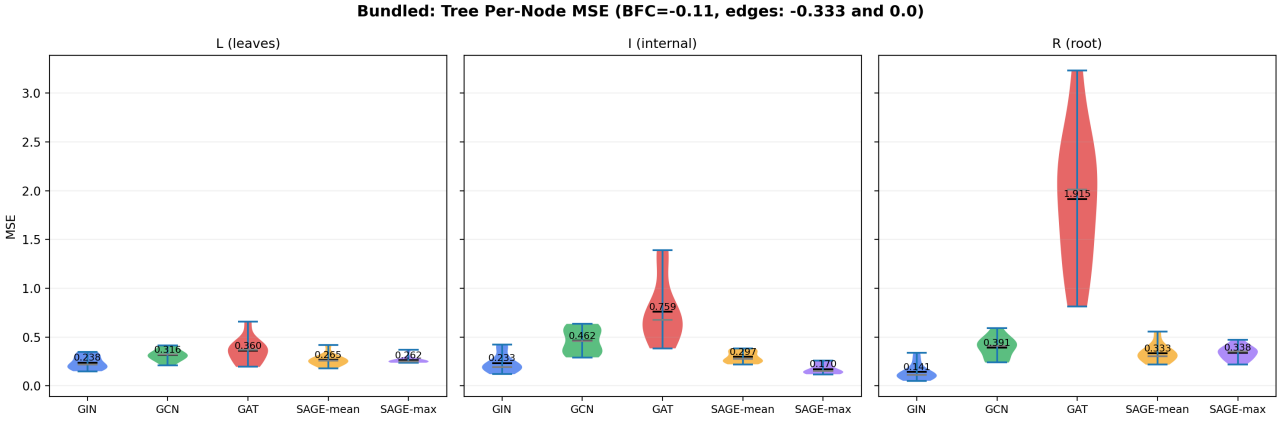


Figure 9. Per-node MSE distribution on the binary tree (bundled setting). Root, internal, and leaf nodes prefer different operators, with up to $14\times$ MSE differences at the root. The setup of the experiment is demonstrated in Figure 4

.

Table 6 contrasts the two experimental regimes used in Section 2.10. The decoupled experiment fixes the message function ψ and updates φ and varies only the aggregation Σ, isolating the causal effect of aggregation. The bundled experiment uses five real GNN operators in which ψ, Σ, and φ all co-vary, examining how full GNN behaviour manifests across geometric regimes.

H. Synthetic Experiments: Task Dependence on the Binary Tree

This appendix presents per-node MSE for the three task assignments referenced in Section 2.10. Geometry and feature distribution are held identical across the three; only the task carried by each node changes. We show leaves (L), internal nodes (I), and the root (R) panels in each figure. In the neutral setting, I carries no task, so the I panel reports only behaviour and is not directly comparable to the trained I in the other two settings; the L and R panels remain the principal point of comparison.

Across the three conditions, Figure 10 shows the operator preference patterns shifting in stability rather than mean. Sum is tight and accurate under reversed and neutral, but exhibits high across-seed variance at R in the original setting. The degree norm is tight at I under the reversed setting but loosens at L and R, whereas in the original setting it becomes tight and accurate at both L and R. Mean continues to fail at the root, irrespective of condition, consistent with the bottleneck

Table 6. Decoupled and bundled formulations. In the decoupled experiment, ψ and φ are fixed and only Σ varies. In the bundled experiment, all three components are determined by the choice of operator. $\mathbf{h}_j$ is the feature vector of neighbour j, $\mathbf{W}$ a learned linear transformation, and $\tilde{d}_i = |\mathcal{N}(i)| + 1$ degree with self-loop.

Variant	**Message ψ**	**Aggregation Σ**	**Update φ**
Decoupled (only Σ varies)			
Sum	$\mathbf{W}\mathbf{h}_j$	$\sum_{j\in\tilde{\mathcal{N}}(i)} \mathbf{h}_j$	MLP
Mean	$\mathbf{W}\mathbf{h}_j$	$\frac{1}{\tilde{d}_i}\sum_{j\in\tilde{\mathcal{N}}(i)} \mathbf{h}_j$	MLP
Degree-norm	$\mathbf{W}\mathbf{h}_j$	$\sum_{j\in\tilde{\mathcal{N}}(i)} \frac{1}{\sqrt{\tilde{d}_i\tilde{d}_j}}\mathbf{h}_j$	MLP
Max	$\mathbf{W}\mathbf{h}_j$	$\max_{j\in\tilde{\mathcal{N}}(i)} \mathbf{h}_j$ (element-wise)	MLP
Bundled (ψ, Σ, φ all co-vary)			
GIN	$\mathbf{h}_j$	sum	$\mathrm{MLP}((1+\epsilon)\mathbf{h}_i + \sum \cdot)$
GCN	$\mathbf{W}\mathbf{h}_j$	symmetric-norm sum	$\sigma(\sum \cdot)$
GAT	$\mathbf{W}\mathbf{h}_j$	attention-weighted	$\sigma(\sum \alpha_{ij} \cdot)$
SAGE-mean	$\mathbf{h}_j$	mean over $\mathcal{N}(i)$	$\sigma(\mathbf{W}[\mathbf{h}_i \,\|\, \cdot])$
SAGE-max	$\mathbf{h}_j$	element-wise max over $\mathcal{N}(i)$	$\sigma(\mathbf{W}[\mathbf{h}_i \,\|\, \cdot])$

pressure the root sits under, regardless of where the task is assigned.

Grid: Per node task dependence. The same probe applied to G_{grid}, where the intermediate node m sits between A and B, shows the analogous task driven shift. In the original assignment (m predicts A), per-node MSE is high at A (1.15) and low at B (0.71); in the reversed assignment (m predicts B), it flips, with A at 0.77 and B at 1.21. The beneficiary is the node whose own prediction target coincides with m's: when m predicts A, it tunes its representation toward A, which is also the target that B must predict, so m supplies B with a representation already aligned to B's task and B's error drops (orig: B at 0.71, A at 1.15). The reverse holds when m predicts B (rev: A at 0.77, B at 1.21). Per-operator MSE values for both conditions are reported in Figures 11a and 11b.

I. Statistical Framework: Full Treatment

This appendix gives the full derivation of the framework introduced in Section 2.6.

I.1. Setup

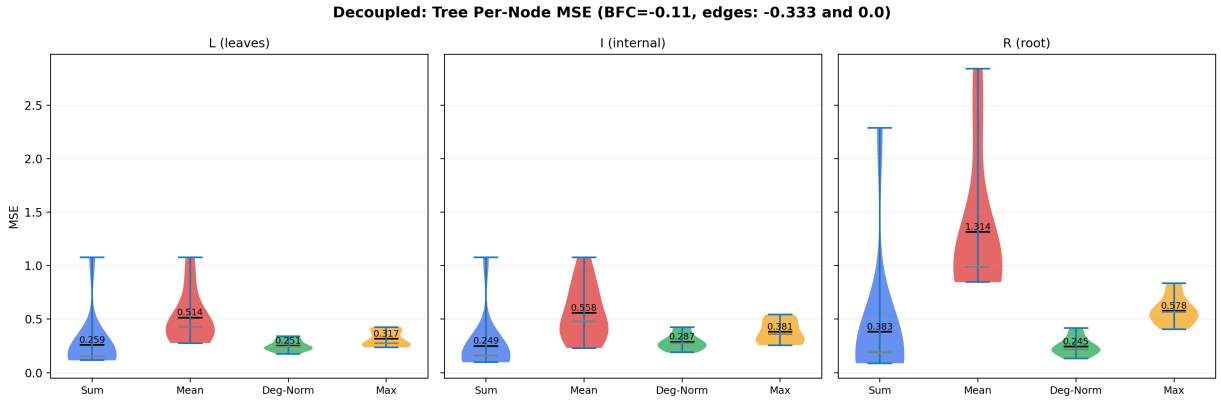
Let $\mathcal{D}$, $\mathcal{SS}$, and $\mathcal{B}$ denote the sets of datasets, operator search spaces, and architectural envelopes considered, and let

$$\mathcal{C} \;\subseteq\; \mathcal{D} \times \mathcal{SS} \times \mathcal{B} \tag{16}$$

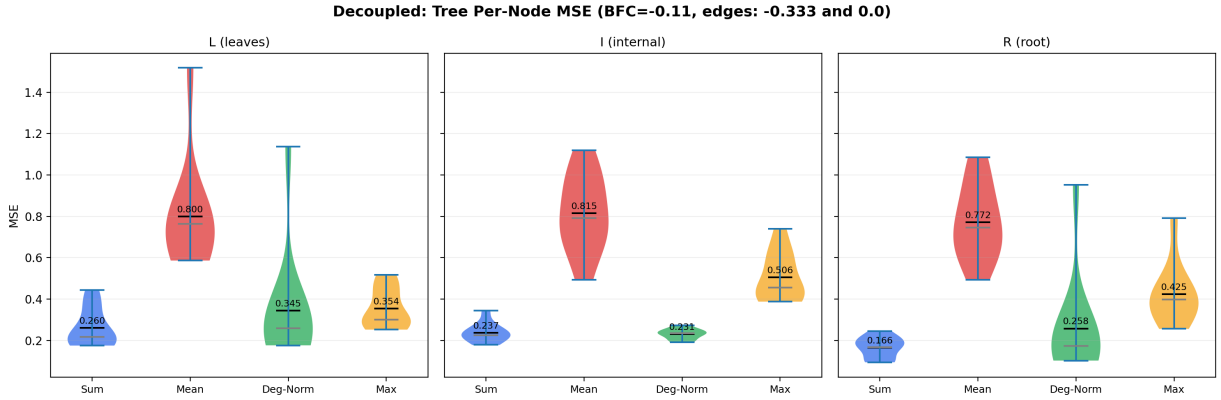
denote the set of configurations actually evaluated. In our experiments $|\mathcal{C}| = N_c = 31$. Each $c = (d, s, b) \in \mathcal{C}$ is a 3-tuple, with depth L_c layers indexed $\ell \in \{0, 1, \ldots, L_c - 1\}$. Total layer count $L_{\text{total}} = \sum_{c\in\mathcal{C}} L_c = 166$. Let $K = |\mathcal{O}|$ denote the number of candidate operators.

I.2. Null Hypothesis

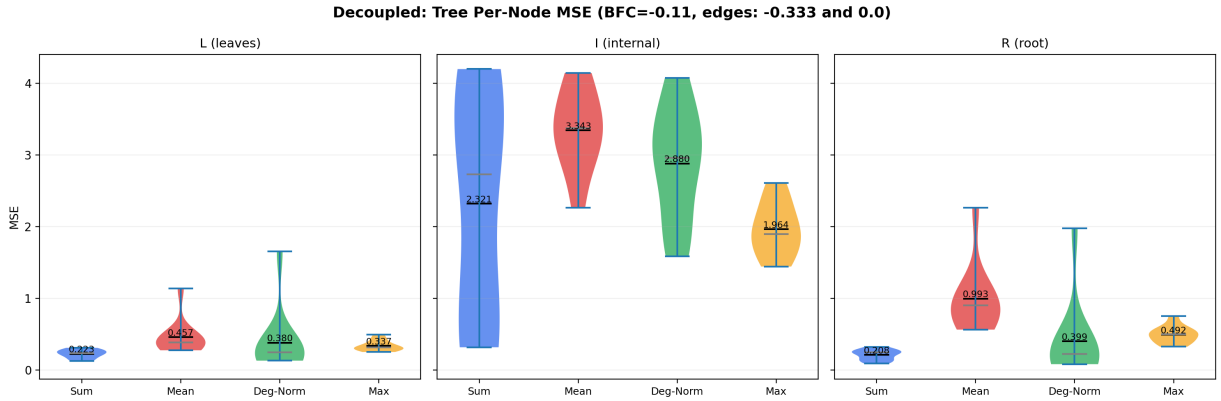
For each layer ℓ in configuration c, the N-seed DARTS ensemble produces per-run winners. Let $c_{\ell,k}$ denote the vote count for operator k at layer ℓ, i.e. the number of seeds whose DARTS run selected operator k at that layer. The null hypothesis H_0



(a) Original (I predicts R).

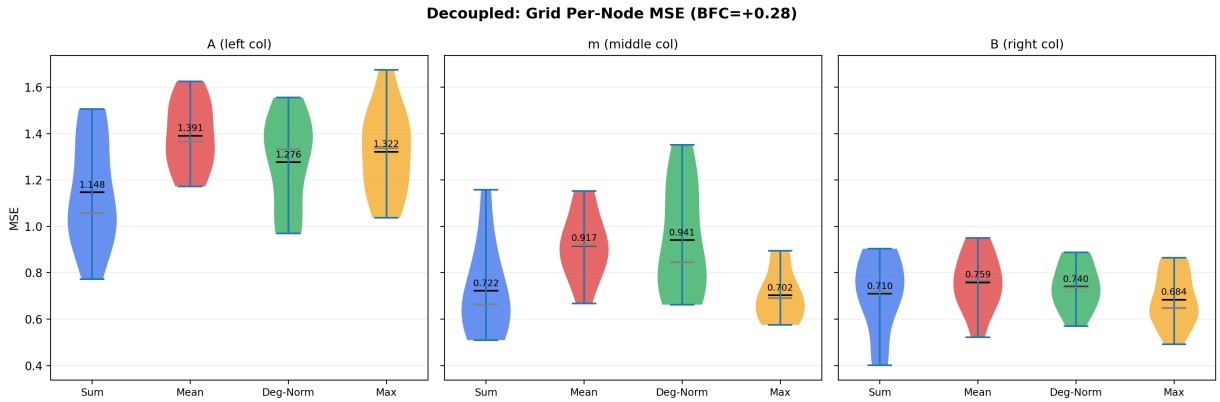


(b) Reversed (I predicts L).

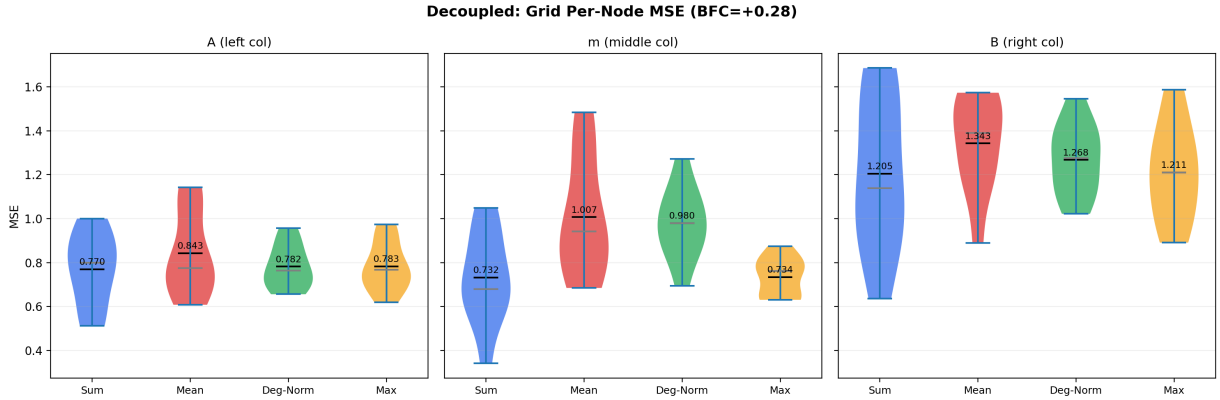


(c) Neutral (I carries no task).

Figure 10. **Per node MSE on G_{tree} under three task assignments.** (a) Sum at R shows mean 0.383 with substantial across seed variance; degree norm is tight and accurate at both L and R. (b) Degree norm is tightest at I (0.231), looser at L (0.345) and R (0.258); sum tight everywhere. (c) The I panel reports relay only behaviour and is excluded from operator preference comparison; sum and degree-norm remain competitive in mean at L and R, while degree-norm loosens relative to its original setting profile. Mean fails at R across all conditions, indicating that the root bottleneck imposes an operator constraint that holds regardless of how task pressure is distributed across the tree.



(a) Original assignment (m predicts A). High MSE at A, low at B.



(b) Reversed assignment (m predicts B). High MSE at B, low at A.

Figure 11. **Per node MSE on G_{grid} under two task assignments.** The node sharing m's prediction target benefits with lower error, while the other node carries the stress.

is that DARTS selects operators at random, with no real preference, and that each operator wins each layer independently with probability $p_0 = 1/K$. Under H_0, the vote vector $(c_{\ell,1}, \ldots, c_{\ell,K})$ follows

$$(c_{\ell,1}, \ldots, c_{\ell,K}) \sim \text{Multinomial}(N, p_0), \qquad p_0 = (1/K, \ldots, 1/K). \tag{17}$$

This null tests whether the per-layer choice is a real signal versus optimisation noise. A complementary test of layer-dependence of operator preference — a direct test of layer-wise heterogeneity is presented in Appendix J at higher seed count ($N = 38$ per Peptides task, pooled $N = 76$ across both tasks; $p = 6.69 \times 10^{-17}$ pooled).

I.3. Aspects of the Preference Distribution

A test T on $(c_{\ell,1}, \ldots, c_{\ell,K})$ is characterised by the *aspect* $\mathcal{A}_T$ it probes, defined as the set of departures from the uniform multinomial that T is sensitive to. Two tests T_i, T_j are *complementary* when their aspects are pairwise non-containing:

$$\mathcal{A}_{T_i} \not\subseteq \mathcal{A}_{T_j} \quad \text{and} \quad \mathcal{A}_{T_j} \not\subseteq \mathcal{A}_{T_i}. \tag{18}$$

Joint rejection of H_0 by every test in $\mathcal{T} = \{T_1, \ldots, T_J\}$ rules out alternatives in the union $\bigcup_j \mathcal{A}_{T_j}$, strictly larger than the alternatives ruled out by any single test alone.

I.4. Derivation of the Null Baseline

For each test T_j at layer ℓ in configuration c, the within-configuration Bonferroni-corrected p-value is $p_j^{(c,\ell)} = L_c \cdot p_j^{\text{raw}}$. Let $R_j = \{(c, \ell) : p_j^{(c,\ell)} < \alpha\}$ and define the indicator $X_{c,\ell}^{(j)} = \mathbf{1}[p_j^{(c,\ell)} < \alpha]$, so $|R_j| = \sum_{c,\ell} X_{c,\ell}^{(j)}$.

Under H_0, valid raw p-values are super-uniform, so for any $t \in [0, 1]$,

$$\mathbb{P}_{H_0}\left[p_j^{\text{raw}} < t\right] \;\le\; t. \tag{19}$$

Consequently,

$$\mathbb{P}_{H_0}\left[X_{c,\ell}^{(j)} = 1\right] \;=\; \mathbb{P}_{H_0}\left[p_j^{\text{raw}} < \frac{\alpha}{L_c}\right] \;\le\; \frac{\alpha}{L_c}. \tag{20}$$

By linearity of expectation,

$$\begin{aligned}\mathbb{E}_{H_0}[|R_j|] &= \sum_{c\in\mathcal{C}} \sum_{\ell=0}^{L_c-1} \mathbb{P}_{H_0}\left[X_{c,\ell}^{(j)} = 1\right] \\ &\le \sum_{c\in\mathcal{C}} L_c \cdot \frac{\alpha}{L_c} \;=\; \alpha\, N_c.\end{aligned} \tag{21}$$

Each configuration thus contributes at most α to the expected number of false layer-level rejections, regardless of its depth, and the bound grows only with N_c.

For $\alpha = 0.05$ and $N_c = 31$,

$$\mathbb{E}_{H_0}[|R_j|] \;\le\; \alpha \cdot N_c \;=\; 1.55. \tag{22}$$

Hence 1.55 is the Bonferroni null upper bound on the expected number of false-positive layers for each test T_j. This holds without requiring equal layer counts across configurations or independence among the layer-wise tests.

I.5. Joint-Rejection Criterion

The per-layer signal is declared real when, for every $T_j \in \mathcal{T}$, the observed rejection count substantially exceeds the null baseline:

$$|R_j| \;\gg\; \alpha \cdot N_c \quad \text{for all } j = 1, \ldots, J, \tag{23}$$

where $\alpha \cdot N_c = 1.55$ is the Bonferroni null upper bound for $\alpha = 0.05$, $N_c = 31$. A signal satisfying this criterion cannot be explained by chance-level operator selection on any of the J aspects probed, and complementarity (Equation 18) ensures the criterion covers genuinely distinct departures from H_0.

I.6. Instantiation: Three Complementary Tests

We instantiate $\mathcal{T}$ with $J = 3$ complementary tests.

T_1**, per-layer chi-square goodness-of-fit.** Tests whether the full vote distribution at layer ℓ deviates from random uniform selection:

$$\chi_\ell^2 \;=\; \sum_{k=1}^{K} \frac{(c_{\ell,k} - N/K)^2}{N/K}, \qquad \chi_\ell^2 \;\sim\; \chi_{K-1}^2 \;\text{ under } H_0. \tag{24}$$

Aspect $\mathcal{A}_{T_1}$: any deviation in the full vote distribution from uniform.

At $N = 10$ the expected per-operator count under H_0 is $N/K \in \{2.5, 3.3\}$, modest for the asymptotic χ^2 approximation; we therefore corroborate every chi-square result with the exact binomial (T_2) and Friedman (T_3) tests, which do not rely on this approximation, and note that observed rejection counts far exceed the null bound.

T_2**, per-layer exact binomial.** Tests whether the winning operator's vote count $c_\ell^* = \max_k c_{\ell,k}$ exceeds what random selection would produce:

$$p_\ell^{\text{bin}} \;=\; \min\left\{1,\; K \sum_{j=c_\ell^*}^{N} \binom{N}{j} p_0^j \,(1-p_0)^{N-j}\right\}, \tag{25}$$

where the factor of K corrects for the fact that the winner is identified post-hoc rather than pre-specified. Unlike chi-square, this is exact and does not require asymptotic approximation. Aspect $\mathcal{A}_{T_2}$: concentration of votes on a single operator.

T_3, per-layer Friedman test. Operates on the architecture parameters $\alpha_{i,\ell,k}$ from each seed i rather than on the discrete winners. Within layer ℓ, operators are ranked by their α-strength in each seed; let $\bar{r}_{\ell,k}$ denote the mean rank of operator k across seeds. The Friedman statistic is

$$Q_\ell \;=\; \frac{12N}{K(K+1)} \sum_{k=1}^{K} \left(\bar{r}_{\ell,k} - \frac{K+1}{2} \right)^2, \qquad Q_\ell \;\sim\; \chi^2_{K-1} \text{ under } H_0. \tag{26}$$

Friedman uses the full ordering rather than only the winner, making it sensitive to consistent ranking patterns even when no single operator dominates the discrete vote. Aspect $\mathcal{A}_{T_3}$: rank consistency in $\boldsymbol{\alpha}$ even when discrete winners are split across operators.

I.7. Verification of Complementarity

We verify pairwise non-containment of $\mathcal{A}_{T_1}, \mathcal{A}_{T_2}, \mathcal{A}_{T_3}$ by exhibiting departures from H_0 detected by some tests but not others.

- A vote distribution $(c_{\ell,1}, \ldots, c_{\ell,K})$ where one operator receives a sharp vote spike while the rest are uniform is detected by T_2 (binomial detects winner concentration) and T_1 (chi-square detects total-distribution deviation), but T_3 may not detect it if non-winner operators have inconsistent α rankings across seeds.
- A distribution where $\boldsymbol{\alpha}$ rankings are consistent across seeds (one operator's mean α dominates) but discrete winners are split (different operators win in different seeds with similar margins) is detected by T_3 but not by T_2.
- A distribution that perturbs the uniform multinomial mildly across all operators without producing a strong winner or consistent ranking is detected by T_1 but may not be detected by T_2 or T_3.

These examples establish that no single $\mathcal{A}_{T_i}$ contains another, satisfying Equation 18.

I.8. Configuration-Level Pooling: Fisher's Combined Test

As a complementary configuration-level statistic, we combine per-layer chi-square p-values within each configuration via Fisher's method:

$$T^F \;=\; -2 \sum_{\ell=0}^{L_c-1} \ln p_\ell^{\chi^2}, \qquad T^F \;\sim\; \chi^2_{2L_c} \text{ under } H_0. \tag{27}$$

We then apply across-configuration Bonferroni correction at the threshold $\alpha/N_c \approx 1.6 \times 10^{-3}$. Configurations rejected at this threshold provide a coarser, per-configuration view of the same evidence. In our experiments, $20/31$ configurations are rejected at this corrected threshold (Table 7).

The $\chi^2_{2L_c}$ calibration assumes the per-layer p-values are independent across layers. Within a configuration, the layers are coupled through the shared trained network (each layer's input depends on the preceding layers), so the per-layer p-values are not strictly independent, and we treat the Fisher statistic as an approximate, descriptive per-configuration aggregate rather than an exact test; the per-layer tests (chi-square, binomial, Friedman) remain the primary evidence.

J. Statistical Robustness at Higher N and Across DARTS Variants: Peptides Case Study

This appendix presents supplementary evidence at higher seed count and across both DARTS variants, on Peptides-func and Peptides-struct (SS1, $L = 6$). It complements the main-paper analysis (Section 2.6) by showing (i) the per-layer preference signal strengthens substantially as N increases, addressing concerns about statistical power at $N = 10$, and (ii) the same preference patterns emerge from first-order and second-order DARTS, ruling out gradient-approximation artefacts. We use 19 seeds $\times$ 2 DARTS variants = 38 runs per task, with 76 runs combined across the two tasks (the same molecular graphs underlie both, so combining is meaningful for studying the underlying preference).

J.1. Preference Counts at $N = 76$

Table 8 shows the combined preference counts $C_\ell(o)$ at $N = 76$.

J.2. Statistical Tests at Higher N

We apply the same chi-square independence test (combined across layers as in Appendix I) and the per-layer exact binomial test against the no-preference null $p_0 = 1/|\mathcal{O}| = 0.25$, with Bonferroni correction across $L = 6$ layers.

Table 7. Configuration-level statistical test results across all 31 (dataset, SS, backbone) configurations. **Backbone**: Ent = Entangled-GT, Dis = Disentangled-GT, Pure = Pure GNN. **Fisher** p: combined p-value across per-layer chi-square tests. χ^2, **Bin**, **Fr**: number of layers (out of L) rejecting at the within-config Bonferroni threshold for chi-square goodness-of-fit, exact-binomial (post-hoc winner-corrected), and Friedman tests. **Nom**: rejects Fisher at $\alpha = 0.05$. **Bonf**: rejects Fisher at across-config Bonferroni $\alpha/31 \approx 1.6 \times 10^{-3}$. **Seeds**: $N = 10$.

Dataset	**SS**	**Bckbn**	L	**Fisher** p	χ^2	**Bin**	**Fr**	**Nom**	**Bonf**
Pep-struct	SS2	Ent	2	0.015	1/2	1/2	1/2	✓	
Pep-struct	SS2	Dis	2	5.0×10^{-5}	2/2	2/2	2/2	✓	✓
Pep-struct	SS2	Pure	2	8.8×10^{-5}	1/2	1/2	1/2	✓	✓
Pep-struct	SS1	Ent	2	0.086	0/2	0/2	0/2		
Pep-struct	SS1	Dis	2	2.4×10^{-5}	1/2	1/2	2/2	✓	✓
Pep-struct	SS1	Pure	2	2.7×10^{-3}	1/2	1/2	1/2	✓	
Pep-func	SS2	Ent	2	4.6×10^{-4}	1/2	1/2	2/2	✓	✓
Pep-func	SS2	Dis	2	9.1×10^{-4}	1/2	1/2	1/2	✓	✓
Pep-func	SS2	Pure	2	6.6×10^{-5}	2/2	2/2	2/2	✓	✓
Pep-func	SS1	Ent	2	3.0×10^{-4}	1/2	1/2	1/2	✓	✓
Pep-func	SS1	Dis	2	5.6×10^{-4}	1/2	1/2	1/2	✓	✓
Pep-func	SS1	Pure	2	4.7×10^{-5}	1/2	1/2	1/2	✓	✓
PATTERN	SS2	Ent	6	3.5×10^{-7}	4/6	2/6	3/6	✓	✓
PATTERN	SS2	Dis	6	2.6×10^{-4}	1/6	1/6	3/6	✓	✓
PATTERN	SS2	Pure	6	3.2×10^{-4}	1/6	0/6	1/6	✓	✓
PATTERN	SS1	Dis	6	4.4×10^{-3}	0/6	0/6	1/6	✓	
PATTERN	SS1	Pure	6	0.021	1/6	1/6	2/6	✓	
CLUSTER	SS2	Ent	16	2.5×10^{-29}	10/16	10/16	12/16	✓	✓
CLUSTER	SS1	Ent	16	1.5×10^{-63}	15/16	15/16	15/16	✓	✓
MolHIV	SS2	Ent	8	0.211	0/8	0/8	0/8		
MolHIV	SS2	Dis	8	0.211	0/8	0/8	0/8		
MolHIV	SS2	Pure	8	0.070	0/8	0/8	1/8		
MolHIV	SS1	Ent	8	2.3×10^{-5}	2/8	2/8	1/8	✓	✓
MolHIV	SS1	Dis	8	7.6×10^{-8}	2/8	1/8	1/8	✓	✓
MolHIV	SS1	Pure	8	0.644	0/8	0/8	0/8		
MolPCBA	SS2	Ent	4	1.1×10^{-3}	1/4	1/4	2/4	✓	✓
MolPCBA	SS1	Ent	4	0.543	0/4	0/4	0/4		
MalNet-T	SS2	Ent	4	4.1×10^{-8}	3/4	3/4	2/4	✓	✓
MalNet-T	SS1	Ent	4	0.379	0/4	0/4	0/4		
ZINC	SS2	Ent	8	7.8×10^{-4}	1/8	0/8	0/8	✓	✓
ZINC	SS1	Ent	8	1.7×10^{-7}	2/8	2/8	4/8	✓	✓
				Total layers rejecting (of 166):	**56**	**51**	**63**		
							Total configs rejecting (of 31):	**24**	**20**

Table 8. Combined preference counts $C_\ell(o)$ over 76 runs (both Peptides tasks pooled, same SS1 search space, best-epoch $\boldsymbol{\alpha}$). Bold = plurality winner.

Layer	**GIN**	**SAGE**	**GCN**	**GAT**
$\ell = 0$	11	11	19	**35**
$\ell = 1$	18	3	5	**50**
$\ell = 2$	**33**	1	9	**33**
$\ell = 3$	**46**	2	10	18
$\ell = 4$	**47**	2	17	10
$\ell = 5$	**46**	2	14	14

Table 9. Chi-square test of independence (H_0: $C_\ell(o)$ is independent of ℓ) on Peptides SS1 at higher N.

Configuration	χ^2	*df*	p**-value**
Peptides-func ($N = 38$)	98.10	15	2.98×10^{-14}
Peptides-struct ($N = 38$)	40.72	15	3.52×10^{-4}
Combined ($N = 76$)	111.99	15	6.69×10^{-17}

Table 10. Per-layer exact binomial tests on the combined $N = 76$ Peptides data, Bonferroni-corrected across $L = 6$ layers (without the post-hoc K winner correction of Appendix I, equation 25, since here the winner identity is reported alongside). All six layers reject the no-preference null. Even under the additional post-hoc $K = 4$ winner correction (multiplying each p-value by K, as in Equation 25), all six layers continue to reject the no-preference null (weakest corrected $p = 8.36 \times 10^{-3} < 0.05$).

Layer	**Winner (count)**	p_{Bonf}
$\ell = 0$	GAT (35/76)	3.31×10^{-4}
$\ell = 1$	GAT (50/76)	4.94×10^{-13}
$\ell = 2$	GIN (33/76)	2.09×10^{-3}
$\ell = 3$	GIN (46/76)	3.54×10^{-10}
$\ell = 4$	GIN (47/76)	7.42×10^{-11}
$\ell = 5$	GIN (46/76)	3.54×10^{-10}

The combined p-value of 6.69×10^{-17} rejects layer independence with overwhelming confidence, and each layer's plurality winner is individually significant after multiple-testing correction. The signal observed at $N = 10$ in the main paper persists and strengthens at $N = 76$.

J.3. Inter-Variant Agreement (First-order vs Second-order DARTS)

Table 11 compares the layer-wise winners (operators with highest mean $\bar{\alpha}_o^{(\ell)}$ across 19 seeds) under first-order and second-order DARTS, on each task.

Table 11. Per-layer winning operator (highest mean $\bar{\alpha}_o^{(\ell)}$) on Peptides SS1 under first-order and second-order DARTS, 19 seeds each. All 12-layer task pairs agree across variants.

	Peptides-func		**Peptides-struct**	
Layer	**1st-order**	**2nd-order**	**1st-order**	**2nd-order**
$\ell = 0$	GAT	GAT	GAT	GAT
$\ell = 1$	GAT	GAT	GAT	GAT
$\ell = 2$	GAT	GAT	GIN	GIN
$\ell = 3$	GIN	GIN	GIN	GIN
$\ell = 4$	GIN	GIN	GIN	GIN
$\ell = 5$	GIN	GIN	GIN	GIN

The winning operator agrees at all 12 layer-task pairs across variants. The second-order signal is compressed slightly closer to uniform (consistent with the Hessian-correction producing a more regularised $\boldsymbol{\alpha}$ trajectory), but the directional structure i.e. which operator is preferred at each layer is identical.

J.4. Summary

The per-layer preference signal strengthens dramatically at higher N ($p = 6.69 \times 10^{-17}$ at $N = 76$, every layer rejecting the no-preference null at Bonferroni-corrected significance), and the pattern is invariant to first-order vs second-order DARTS. Together, these confirm that the layer-wise preferences reported at $N = 10$ in the main paper are a robust property of the data and the search procedure, rather than artefacts of low statistical power or a specific gradient approximation.

K. Rationale for the Three Architectural Envelopes

The three architectural envelopes used in this work are visually described in 12

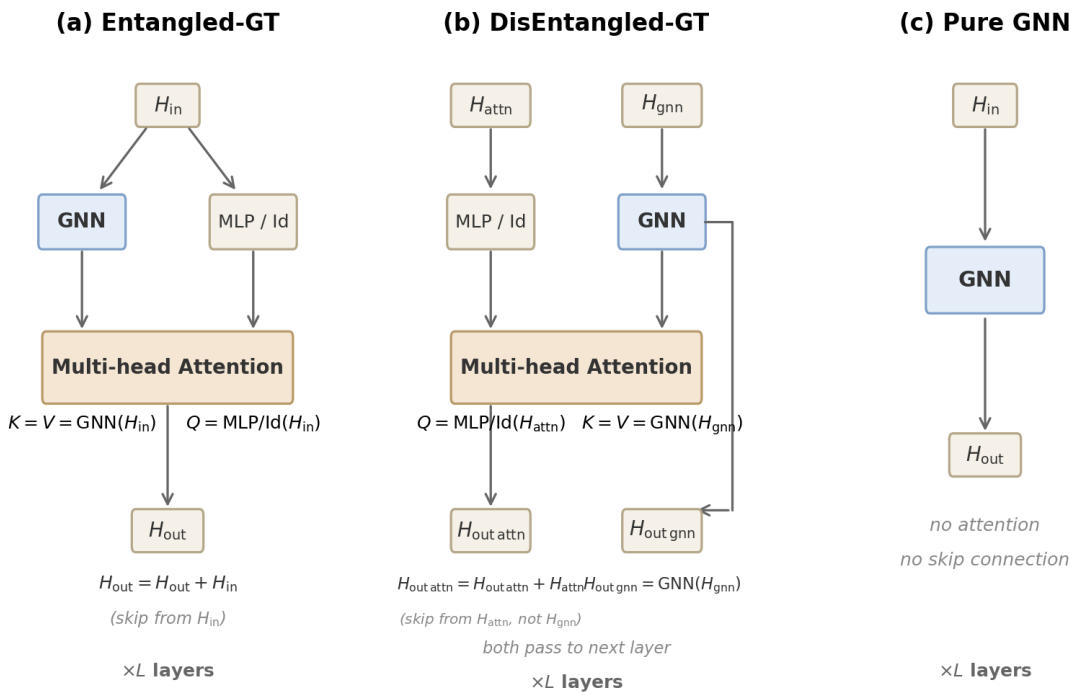


Figure 12. The three architectural envelopes used in this work. Entangled-GT: GNN supplies keys and values to the attention block, queries from node features. DisEntangled-GT: two parallel streams, the attention stream supplies queries and the GNN stream supplies keys and values. Pure GNN: stacked GNN layers with no attention.

The three envelopes evaluated in this paper — Entangled-GT (adopted from (Chakraborty & Minary, 2025)), DisEntangled-GT, and Pure GNN were chosen to span a controlled spectrum of local-global mixing, not because we propose them as new architectures.

Entangled-GT. As the name suggests, the per-layer GNN is entangled with global attention: the GNN-derived keys and values pass through the attention block, so its output is mixed with the global signal at every layer. Subsequent layers receive this already-mixed representation as input. The per-layer GNN therefore operates not on a pure local signal but on a signal that has been globally mixed at every prior depth.

DisEntangled-GT. Two parallel streams flow through each layer. The attention stream supplies queries, and the GNN stream supplies keys and values, but the residual is on the attention stream only — meaning the GNN stream is never written back into by attention. Even at the final layer, the GNN sees only locally derived input, while the attention output flows independently. This isolates the per-layer GNN as a fully local operator under the envelope of a global attention network. The interesting question this raises is: does layer-wise heterogeneity persist when the per-layer GNN is held strictly local—and, if so, does the heterogeneity pattern align with the Entangled-GT consensus or diverge from it?

Pure GNN. Stacked GNN layers with no attention. This is the conventional GNN backbone — the standard practice in much of the GNN literature and serves as the no-attention boundary of the envelope spectrum.

Why not GPS? GPS (Rampášek et al., 2022) runs a fully parallel MPNN-attention design with both branches mixed together at every layer. This was not included in our envelope study for two reasons: (i) the parallel-path mixing makes it harder to attribute the per-layer GNN selection to a single causally controlled axis; (ii) the original GPS supplementary results showed that on several benchmarks, removing the GPS's GNN component (attention-only ablation) substantially

degraded performance, suggesting the optimisation pressure is concentrated on one path, complicating the diagnostic interpretation of per-layer GNN preferences. Whether layer-wise heterogeneity persists under GPS-style fully parallel mixing is an open question that this paper does not address, but that we view as a natural follow-up direction.

Scope. The aim of this study is not to propose a new NAS protocol or a new Graph Transformer architecture. The aim is to test whether the per-layer heterogeneity phenomenon holds across functionally distinct architectural envelopes — when the per-layer GNN is mixed with attention at every depth (Entangled-GT), when it is held strictly local under attention (DisEntangled-GT), and when no attention is present at all (Pure GNN). The competitive-to-SOTA results reported as a by-product (Tables 4 and 5) demonstrate that the resulting consensus architectures are practically usable even under robust, reproducible conditions, but the central contribution is diagnostic, not architectural.

L. Datasets and Hyperparameters

L.1. Datasets

We evaluate across eight benchmarks spanning molecular property prediction (ZINC with 12,000 graphs averaging 23.2 nodes; OGBG-MolHIV with 41,127 graphs averaging 25.5 nodes; OGBG-MolPCBA with 437,929 graphs averaging 26.0 nodes), peptide property prediction (Peptides-Struct and Peptides-Func, 15,535 graphs each averaging 150.9 nodes), node-level synthetic classification (PATTERN with 14,000 graphs averaging 118.9 nodes; CLUSTER with 12,000 graphs averaging 117.2 nodes), and malware call-graph classification (MalNet-Tiny with 5,000 graphs averaging 1,410.3 nodes). Tasks span graph-level regression (ZINC, Peptides-Struct), graph-level binary and multi-task classification (MolHIV, MolPCBA, Peptides-Func, MalNet-Tiny), and node-level binary and multi-class classification (PATTERN, CLUSTER). We follow the official train/val/test splits provided with each benchmark.

L.2. Hyperparameters

We adopt the GPS (Rampášek et al., 2022) configuration as the starting point for all datasets (positional encoding, hidden dimension, heads, dropout, and optimiser settings), adjusting only the layer count to maintain a comparable parameter budget to Standard-GT. The full per-dataset hyperparameter configuration follows the GPS (Rampášek et al., 2022) configuration, with backbone hyperparameter choices held constant, leaving the per-layer GNN selection as the only varying component. Tables 12 and 13 summarise these settings, grouped into architecture parameters, DARTS parameters, training parameters, graph-specific parameters, and computational resources.

Q-projection is the identity map by default, with an MLP applied to Peptides when features benefit from a learned query projection. The edge residual weight $\sigma_g(\gamma^{(\ell)})$ is initialised so that $\sigma_g(\gamma^{(\ell)}) = 0.1$ at the start of training; "NA" in the Edge residual weight row indicates the edge residual stream was disabled for that dataset.

For DARTS search, the training data is split 60/40 into search-train and search-validation subsets. Architecture parameters are updated on the search-validation subset using a separate Adam optimiser with learning rate 4×10^{-4}. The discrete architecture is then retrained from scratch on the full training set using the training hyperparameters in Tables 12 and 13. The *Total DARTS time* reported in the tables is the wall-clock time for a single DARTS search run; the ensemble protocol (Section 2.3) runs $N = 10$ such searches per (dataset, search space, backbone) configuration, so the total ensemble DARTS time is approximately $10\times$ the per-seed value. *Discrete phase time* is reported as per-epoch / total wall-clock for the discrete retraining phase.

L.3. Parameter Counts

Table 14 compares parameter counts across Vanilla-GT, GPS, and the hybrid-consensus architecture on the Entangled-GT backbone, separated by attention type. The hybrid-consensus parameter counts are slightly higher than Vanilla-GT on most datasets due to the per-layer GNN operators introduced into the attention path; on MolHIV, MolPCBA (sparse), Peptides-struct, and Peptides-func, the hybrid consensus achieves a lower parameter count than at least one strong baseline.

L.4. Hardware and Compute

All experiments were run on Oxford ARC HTC clusters, using a mix of NVIDIA GPUs with 16 GB, 32 GB, and 48 GB of memory depending on availability and the dataset's memory footprint. All runs used a single GPU per job. Total compute consumed across the eight datasets, two search spaces, three backbones, and the $N = 10$ ensemble protocol is approximately

Table 12. Hyperparameters across datasets (Part 1).

Hyperparameters	**ZINC**	**MolHIV**	**MolPCBA**	**MalNet**
Architecture Parameters				
Number of layers	8	8	4	4
Hidden dim	64	64	384	64
Attention type	Sparse	Sparse	Sparse	Sparse
Heads	4	4	4	4
Dropout	0	0.05	0.2	0
Attention dropout	0.5	0.5	0.5	0.5
Edge residual weight	0.1	NA	NA	0.1
Q-projection	Identity	Identity	Identity	Identity
DARTS Parameters				
DARTS epochs	120	30	30	30
DARTS splits ratio	0.6	0.6	0.6	0.6
Architecture LR	4.00E-04	4.00E-04	4.00E-04	4.00E-04
Grad clip	5	5	5	5
Training Parameters				
LR reduce factor	0.5	0.5	0.5	0.5
LR schedule patience	10	10	10	10
Min learning rate	1.00E-06	1.00E-06	1.00E-06	1.00E-06
Initial learning rate	0.0025	0.0025	0.0025	0.0025
Weight decay	3.00E-04	3.00E-04	3.00E-04	3.00E-04
Epochs	2000	100	100	100
Warmup epochs	50	5	5	10
Graph-specific Parameters				
Graph pooling	sum	mean	mean	max
Positional encoding	RWSE-20	RWSE-16	RWSE-16	None
PE dim	28	16	20	NA
PE encoder	linear	linear	linear	NA
Computational Resources				
DARTS parameters	641,989	583,017	10,186,472	327,893
Discrete parameters	412,397	344,929	5,707,484	208,329
Total DARTS time (per seed)	4490s	2777s	4441s	630s
Discrete phase time	26s/15h	63s/2.37h	158s/5h	16s/0.645h

Table 13. Hyperparameters across datasets (Part 2).

Hyperparameters	**PATTERN**	**CLUSTER**	**Pep-Func**	**Pep-Struct**
Architecture Parameters				
Number of layers	6	16	2	2
Hidden dim	64	48	120	120
Attention type	Sparse	Sparse	Dense	Dense
Heads	4	8	4	4
Dropout	0	0	0	0
Attention dropout	0.5	0.5	0.5	0.5
Edge residual weight	0.1	0.1	NA	NA
Q-projection	Identity	Identity	MLP	MLP
DARTS Parameters				
DARTS epochs	20	25	50	50
DARTS splits ratio	0.6	0.6	0.6	0.6
Architecture LR	4.00E-04	4.00E-04	4.00E-04	4.00E-04
Grad clip	5	5	5	5
Training Parameters				
LR reduce factor	0.5	0.5	0.5	0.5
LR schedule patience	10	10	10	10
Min learning rate	1.00E-06	1.00E-06	1.00E-06	1.00E-06
Initial learning rate	0.0025	0.0025	0.0025	0.0025
Weight decay	3.00E-04	3.00E-04	3.00E-04	3.00E-04
Epochs	100	100	200	200
Warmup epochs	5	5	0	0
Graph-specific Parameters				
Graph pooling	NA	NA	mean	mean
Positional encoding	LapPE-16	LapPE-10	LapPE-10	LapPE-10
PE dim	16	16	16	16
PE encoder	DeepSet	DeepSet	DeepSet	DeepSet
Computational Resources				
DARTS parameters	487,369	728,678	604,817	604,696
Discrete parameters	338,775	511,250	426,341	448,240
Total DARTS time (per seed)	1320.45s	3454.26s	409s	411.54s
Discrete phase time	58s/1.87h	97.4s/3.88h	12.49s/0.775h	12.5s/0.845h

Table 14. Comprehensive parameter comparison across all models and datasets. Best (lowest) parameter count is highlighted in **bold**.

Model	Attention	ZINC	MolHIV	MolPCBA	MalNet	PATTERN	CLUSTER	Pep-Struct	Pep-Func
Dense Attention									
Vanilla-GT	Dense	**356,261**	451,793	**8,406,236**	503,749	**302,385**	**385,126**	429,611	429,490
GPS	Dense	423,717	558,625	9,744,496	527,237	337,201	502,054	504,459	504,362
Hybrid (Ours)	Dense	441,751	**398,289**	8,557,152	NA	361,591	507,206	**426,341**	**448,240**
Sparse Attention									
Vanilla-GT	Sparse	**356,261**	451,793	8,406,236	NA	**302,385**	**338,086**	412,397	429,490
Hybrid (Ours)	Sparse	428,333	**344,929**	**5,707,484**	208,329	338,775	511,250	**397,301**	**408,160**

Table 15. Full SOTA comparison on ZINC, MalNet-Tiny, PATTERN, and CLUSTER. Type indicates architecture family: GNN (graph neural network), GT-NAS (NAS-based Graph Transformer), GT (Graph Transformer).

Type	Model	ZINC MAE ↓	MalNet-Tiny Acc ↑	PATTERN Acc ↑	CLUSTER Acc ↑
GNN	GCN (Kipf & Welling, 2017)	0.367±0.011	—	71.892±0.334	68.498±0.976
GNN	GIN (Xu et al., 2019)	0.526±0.051	—	85.387±0.136	64.716±1.553
GNN	GAT (Veličković et al., 2018)	0.384±0.007	—	78.271±0.186	70.587±0.447
GNN	GatedGCN (Bresson & Laurent, 2017)	0.282±0.015	0.9223±0.0065	85.568±0.088	73.840±0.326
GNN	GatedGCN-LSPE (Dwivedi et al., 2022a)	0.090±0.001	—	—	—
GNN	PNA (Corso et al., 2020)	0.188±0.004	—	—	—
GNN	DGN (Beaini et al., 2021)	0.168±0.003	—	86.680±0.034	—
GNN	GSN (Bouritsas et al., 2023)	0.101±0.010	—	—	—
GNN	CIN (Bodnar et al., 2021)	0.079±0.006	—	—	—
GNN	CRaWl (Tönshoff et al., 2023)	0.085±0.004	—	—	—
GNN	GIN-AK+ (Zhao et al., 2022)	0.080±0.001	—	<u>86.850±0.057</u>	—
GT-NAS	EGTAS (Wang et al., 2025a)	0.075±0.073	—	86.742±0.053	**79.236±0.215**
GT-NAS	UGAS (Song et al., 2025)	—	—	**86.89±0.02**	78.14±0.21
GT	SAN (Kreuzer et al., 2021)	0.139±0.006	—	86.581±0.037	76.691±0.65
GT	Graphormer (Ying et al., 2021)	0.122±0.006	—	—	—
GT	K-Subgraph SAT (Chen et al., 2022)	0.094±0.008	—	86.848±0.037	77.856±0.104
GT	EGT (Hussain et al., 2022)	0.108±0.009	—	86.821±0.020	<u>79.232±0.348</u>
GT	GPS (Rampášek et al., 2022)	**0.070±0.004**	**0.9264±0.0078**	86.685±0.059	78.016±0.180
Ours	**Hybrid Consensus**	**0.07±0.004**	**0.926±0.006**	86.65±0.003	78.46±0.015

10× the per-seed DARTS times reported in Tables 12 and 13, plus the discrete-phase retraining times. The synthetic experiments and brute-force validation runs were executed on the same cluster; per-configuration brute-force evaluation on the five small-enough settings (Section 3) added approximately 10–30 GPU-hours each, well within budget for the smaller search spaces.

M. Full SOTA Tables

Tables 15 and 16 provide the full per-baseline comparison referenced in the main paper (Tables 4 and 5). Each row is annotated with its architecture type—GNN, GT-NAS, or GT — to make the groupings explicit. **Bold** marks the best result per dataset, <u>underlined</u> marks the second-best.

N. DARTS: Full Formulation

This appendix expands on the DARTS framework summarised in Section 2.1, covering continuous relaxation, the bilevel optimisation problem, our convergence and best-epoch extraction protocol, and known DARTS failure modes, reason for

Table 16. Full SOTA comparison on MolHIV, MolPCBA, Peptides-Func, and Peptides-Struct.

Type	Model	MolHIV AUROC ↑	MolPCBA AP ↑	Pep-Func AP ↑	Pep-Struct MAE ↓
GNN	GCN (Kipf & Welling, 2017)	—	—	0.5930±0.002	0.3496±0.0013
GNN	GINE (Wang et al., 2022)	—	—	0.5498±0.0079	0.3547±0.0045
GNN	GCN+VN (Rampášek et al., 2022)	0.7599±0.0119	0.2424±0.0034	—	—
GNN	GIN+VN (Rampášek et al., 2022)	0.7707±0.0149	0.2703±0.0023	—	—
GNN	GatedGCN (Bresson & Laurent, 2017)	—	—	0.5864±0.0077	0.3420±0.0013
GNN	GatedGCN-LSPE (Dwivedi et al., 2022a)	—	0.267±0.002	—	—
GNN	PNA (Corso et al., 2020)	0.7905±0.0132	0.2838±0.0035	—	—
GNN	DeeperGCN (Rampášek et al., 2022)	0.7858±0.0117	0.2781±0.0038	—	—
GNN	DGN (Beaini et al., 2021)	0.7970±0.0097	0.2885±0.0030	—	—
GNN	GSN-directional (Bouritsas et al., 2023)	0.8039±0.0090	—	—	—
GNN	CIN (Bodnar et al., 2021)	**0.8094±0.0057**	—	—	—
GNN	CRaWl (Tönshoff et al., 2023)	—	**0.2986±0.0025**	—	—
GNN	GIN-AK+ (Zhao et al., 2022)	0.7961±0.0119	<u>0.2930±0.0044</u>	—	—
GNN	ExpC (Yang et al., 2020)	0.7799±0.0082	0.2342±0.0029	—	—
GT-NAS	AutoGT (Zhang et al., 2023)	0.7495±0.0102	—	—	—
GT-NAS	EGTAS (Wang et al., 2025a)	0.7981±0.0117	—	—	—
GT-NAS	UGAS (Song et al., 2025)	<u>0.7997±0.0051</u>	—	**0.667±0.007**	**0.247±0.002**
GT	SAN (Kreuzer et al., 2021)	0.7785±0.2470	0.2765±0.0042	—	—
GT	SAN+LapPE (Rampášek et al., 2022)	—	—	0.6384±0.0121	0.2683±0.0043
GT	GraphTrans (Wu et al., 2021)	—	0.2761±0.0029	—	—
GT	GPS (Rampášek et al., 2022)	0.7880±0.0101	0.2907±0.0028	0.6535±0.0041	0.2500±0.0005
Ours	**Hybrid Consensus**	0.762±0.01	0.2612±0.002	**0.662±0.005**	**0.247±0.0008**

choosing DARTS over other NAS algorithms for this study, along with the mitigations our protocol provides.

N.1. Continuous Relaxation

Let $G = (V, E)$ be an input graph with node feature matrix $H^{(0)} \in \mathbb{R}^{|V|\times d}$. At each layer ℓ, a GNN operator $o \in \mathcal{O}$ updates node embeddings via message passing, $H^{(\ell)} = o(H^{(\ell-1)}, E)$. During search, DARTS (Liu et al., 2019) relaxes the discrete selection of a single operator into a continuous mixture. At each layer ℓ, instead of selecting one operator, the layer computes a weighted combination

$$\bar{H}^{(\ell)} = \sum_{o\in\mathcal{O}} \frac{\exp(\alpha_o^{(\ell)})}{\sum_{o'\in\mathcal{O}} \exp(\alpha_{o'}^{(\ell)})} \cdot o(H^{(\ell-1)}, E), \tag{28}$$

where $\boldsymbol{\alpha}^{(\ell)} = (\alpha_{o_1}^{(\ell)}, \ldots, \alpha_{o_n}^{(\ell)})^\top \in \mathbb{R}^{|\mathcal{O}|}$ is the vector of architecture parameters at layer ℓ. At discretisation, the selected operation at layer ℓ is $o^{(\ell)*} = \arg\max_{o\in\mathcal{O}} \alpha_o^{(\ell)}$.

N.2. Bi-Level Optimisation

We follow the standard DARTS bi-level formulation (Liu et al., 2019): validation loss is minimised over architecture parameters $\boldsymbol{\alpha}$ subject to weights w being optimal on training loss for the chosen $\boldsymbol{\alpha}$,

$$\min_{\boldsymbol{\alpha}} \ \mathcal{L}_{\text{val}}(w^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}; \mathcal{D}_{\text{val}}) \quad \text{s.t.} \quad w^*(\boldsymbol{\alpha}) = \arg\min_{w} \ \mathcal{L}_{\text{train}}(w, \boldsymbol{\alpha}; \mathcal{D}_{\text{train}}). \tag{29}$$

Since $w^*(\boldsymbol{\alpha})$ is intractable to compute exactly, DARTS alternates between training-data updates of w and validation-data updates of $\boldsymbol{\alpha}$ on a $60/40$ split of the training set. We use the second-order variant, in which the architecture gradient is evaluated after a single inner gradient step $w' = w - \xi\nabla_w\mathcal{L}_{\text{train}}(w, \boldsymbol{\alpha})$; Hessian-vector products in the chain rule are evaluated by finite differences as in (Liu et al., 2019). Full derivation is in the original DARTS paper.

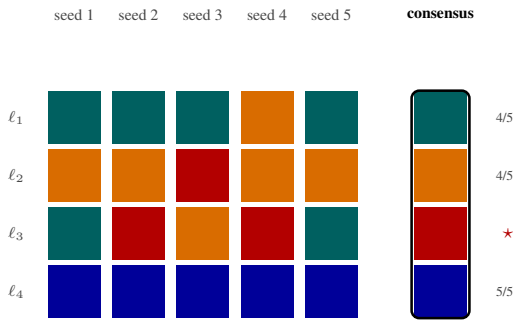


Figure 13. **Ensemble-consensus voting.** Each of N independent DARTS seeds selects an operator at every layer (colours denote operators). A layer resolves by majority vote when $\geq 0.6N$ seeds agree; otherwise it falls back to the operator with the highest mean architecture weight $\bar{\alpha}$ (marked $\star$ at ℓ_3). Per-layer confidence $\kappa_\ell = \max_o c_{\ell,o}/N$ records how decisive each layer is.

N.3. Convergence and Best-Epoch Extraction

DARTS training in our experiments exhibits three convergence regimes:

Converged. Validation loss decreases to a stable minimum. Architecture parameters at the final epoch are reliable.

Mildly overfit. Validation loss reaches a minimum then increases slightly (less than 0.02 above best). We extract $\boldsymbol{\alpha}$ from the latest epoch within 0.02 of the best validation loss, which keeps the extracted $\boldsymbol{\alpha}$ close to the optimal validation regime while allowing values to stabilise.

Diverged. Validation loss increases throughout, or final loss exceeds initial loss. Diverged runs are discarded entirely; the corresponding seed is excluded from the ensemble.

The 0.02 tolerance is small enough to remain within the optimal validation regime, but large enough to allow $\boldsymbol{\alpha}$ values to stabilise between the loss minimum and the extraction epoch.

N.4. Known DARTS Problems and Our Mitigations

DARTS is known to suffer from several pathologies. We discuss each below, along with the mitigations our protocol provides.

Skip-connection dominance. In image-classification NAS, DARTS tends to prefer parameter-free skip connections at the expense of parameterised operators (Liu et al., 2019). *Not applicable in our setting:* all candidate operators in $\mathcal{O}$ are parameterised GNN layers.

Softmax coupling. A weakly preferred operator can be suppressed in favour of a marginally better competitor due to the softmax normalisation in Equation 28. In our experiments, the per-layer α distributions are relatively flat at most layers, suggesting moderate coupling. The hybrid voting/mean rule (Section 2.3) addresses this by falling back to the mean of the architecture parameters in low-vote-concentration layers, where pure voting would be unreliable.

Search–evaluation gap. During search, each layer outputs a weighted mixture of operators; after discretisation, the selected single operator. This gap can degrade the discrete architecture's final performance relative to the continuous mixture. This affects raw performance but does not affect the per-layer preference signal we study, since our scientific observable is $\boldsymbol{\alpha}$ during search rather than discrete-architecture accuracy.

Sensitivity to seeds. A single DARTS run can produce different architectures for different random seeds due to the non-convex bilevel optimisation. This is the principal motivation for the ensemble-consensus protocol (Section 2.3), which aggregates $N = 10$ independent seeds and treats the per-layer winner distribution as the object of study, rather than relying on any single run.

N.5. Why DARTS rather than reinforcement-learning or evolutionary search.

Our object of study is the per-layer operator preference itself, not a single optimised architecture, so the search method must expose that preference as a directly observable internal quantity rather than infer it from outcomes. This is what distinguishes

differentiable search from its alternatives. DARTS optimises architecture parameters $\alpha^{(\ell)}$ by gradient *through* the network, so $\alpha^{(\ell)}$ is the model's own internal architectural state and reads off directly as a continuous per-operator strength at every layer, which is precisely the quantity of our ensemble consensus and the Friedman test operate on. Reinforcement-learning (Gao et al., 2019) and evolutionary (Shi et al., 2022; Liu & Liu, 2023) search treat the trained network as a black box instead i.e. they observe only a scalar reward or fitness at the output and search *over* that signal. Any per-layer preference they yield is therefore inferred indirectly from outcomes, and recovering its distribution demands many full training evaluations per candidate rather than a single shared differentiable pass. DARTS also runs at the cost of one training run per seed, making the multi-seed ensemble protocol tractable across all 31 configurations.

O. Transformer Components

The Transformer components used in Entangled-GT and DisEntangled-GT (Section 2.1, Section 2.5) are standard and used off the shelf, following Vaswani et al. (2017). Each layer applies multi-head attention with M heads and per-head dimension $d_m = d/M$, followed by a position-wise feed-forward network with hidden expansion ratio $r = 2$, with residual connections and layer normalisation after each sublayer. We use either dense (full all-pairs) or sparse (edge-restricted) attention depending on the dataset; the mode affects only the softmax normalisation, not the source of $\mathbf{K}$ and $\mathbf{V}$. The novel design choice in our backbones is which streams supply queries versus which supply keys and values, as described in the main text.

P. Edge Residual Stream

The edge residual stream injects a gated edge term into the post-attention residual, operates independently of the per-layer GNN operator, and therefore does not confound per-layer operator preferences with edge-feature handling.

Q. Per-Head and Granularity Exploration

Before committing to per-layer single-GNN selection in the Entangled-GT backbone, we explored alternative heterogeneity granularities on a controlled, smaller-scale ZINC setup ($M = 4$ heads, single layer, hidden dim 32). The variants studied are:

- **Per-head uniform GNN** (homogeneous baseline): all M heads use the same GNN.
- **Per-head random sampling**: a different GNN is sampled per head at each forward pass.
- **Per-head DARTS, single**: each head's GNN is the DARTS-selected $\arg\max_o \alpha_o^{(m)}$.
- **Per-head uniform mixture**: each head holds a fixed uniform mixture (25%) over all candidate GNNs.
- **Per-head DARTS, weighted mixture**: each head holds a continuous mixture with weights from DARTS architecture parameters $\alpha_o^{(m)}$ via softmax.
- **Shared per-layer-group mixture**: groups of consecutive layers share a single GNN choice.
- **Mixture-of-Experts variant**: a gating network routes each token to one of K candidate GNNs.

Table 17 reports the ZINC MAE for the variants for which we collected complete numerical results. The shared per-layer-group and MoE variants were prototyped but not run to convergence due to gating instabilities and were ruled out qualitatively.

Increasing the granularity below the per-layer scale, where the geometric question is naturally posed, forces compute scaling that does not align with the structural property under study. Per-layer single-GNN selection emerged as the empirically cleanest granularity in pilot experiments on Peptides-func, MalNet-Tiny, and CLUSTER. It produces stable consensus architectures across seeds, exhibits the strongest per-layer statistical signal under the framework of Section 2.6, and admits the cleanest interpretation as a "per-layer aggregation choice." The other granularities either dilute the per-layer signal (per-head) or impose external structure (per-group, MoE) that confounds the diagnostic question.

Table 17. Per-head heterogeneity granularity comparison on ZINC ($M = 4$ heads, single layer, hidden dim 32). MAE averaged across four seeds.

Variant	ZINC MAE ↓	Params
Per-head uniform (homogeneous baseline)		
All heads = GATv2	0.879 ± 0.036	22,721
All heads = GINE	0.636 ± 0.007	35,521
All heads = GatedGCN	0.675 ± 0.016	54,081
Per-head heterogeneous		
Random sampling per head	0.645 ± 0.013	34,005
DARTS, single GNN per head	0.747 ± 0.068	33,163
Uniform mixture per head (25% each)	0.599 ± 0.007	108,585
DARTS, weighted mixture per head	0.618 ± 0.020	108,585

R. Hop-Stratified Over-Squashing Pressure: BFC Formula and Algorithm

R.1. Balanced Forman Curvature

We use Balanced Forman Curvature (BFC) as our scalar geometric descriptor. BFC is an edge-level quantity that combines triangle and 4-cycle contributions with degree information at the edge endpoints. We follow the definition of Topping et al. (Topping et al., 2022) (Definition 1, Equation 3): for an edge (i, j) with degrees d_i, d_j and neighbour sets $\mathcal{N}(i), \mathcal{N}(j)$,

$$\mathrm{Ric}(i,j) = \frac{2}{d_i} + \frac{2}{d_j} - 2 + \frac{2\,|T_{ij}|}{\max(d_i, d_j)} + \frac{|T_{ij}|}{\min(d_i, d_j)} + \frac{|S_i| + |S_j|}{\gamma_{\max} \cdot \max(d_i, d_j)}, \tag{30}$$

where T_{ij} is the set of triangles through edge (i, j), S_i and S_j are sets of qualifying 4-cycle neighbours from each endpoint, and $\gamma_{\max}$ is the maximum number of 4-cycles sharing a common 4-cycle node. The last term is set to zero if no 4-cycles exist; the formula returns 0 if $\min(d_i, d_j) = 1$. We compute BFC per edge for all three synthetic graphs; aggregating to the graph level gives the values reported in Section 2.10 (clique $\mathrm{BFC} = +1.11$ per edge, grid $\mathrm{BFC} \in \{+0.17, +0.33\}$, binary tree $\mathrm{BFC} \in \{-0.33, 0\}$).

R.2. Hop-Stratified Pressure

For source v_0 and hop level ℓ, the hop ring $\Gamma_\ell(v_0) \subseteq E$ collects edges $(a, b) \in E$ with $\max(\mathrm{dist}_G(v_0, a), \mathrm{dist}_G(v_0, b)) = \ell$. Distances are computed by BFS from v_0. The edge (a, b) may be *radial* (one endpoint at distance $\ell - 1$, the other at ℓ) or *lateral* (both at distance ℓ); both are included (Figure 14) . The hop-stratified pressure at source v_0 at hop ℓ is

$$S_{v_0}^{(\ell)} = \sum_{\substack{e \in \Gamma_\ell(v_0) \\ \mathrm{Ric}(e) < 0}} \mathrm{Ric}(e), \tag{31}$$

the graph-level pressure is $\bar{S}^{(\ell)} = \frac{1}{|V|} \sum_v S_v^{(\ell)}$, and dataset-level statistics $\mu_\ell \pm \sigma_\ell$ aggregate across all graphs in the dataset.

R.3. Algorithm

Algorithm 1 Hop-Stratified BFC Pressure

Input: Graph $G = (V, E)$, max hop L
Output: $S_v^{(\ell)}$ for all $v \in V$, $\ell \in \{1, \ldots, L\}$
for *each source* $v_0 \in V$ **do**
  Compute BFS distances $\{\mathrm{dist}_G(v_0, v)\}$ **for** $\ell = 1, \ldots, L$ **do**
    $\Gamma_\ell(v_0) \leftarrow$ edges with max-endpoint-distance $= \ell$ $S_{v_0}^{(\ell)} \leftarrow \sum_{e \in \Gamma_\ell(v_0), \mathrm{Ric}(e) < 0} \mathrm{Ric}(e)$
  end
end
return $S_v^{(\ell)}$ *for all* v, ℓ

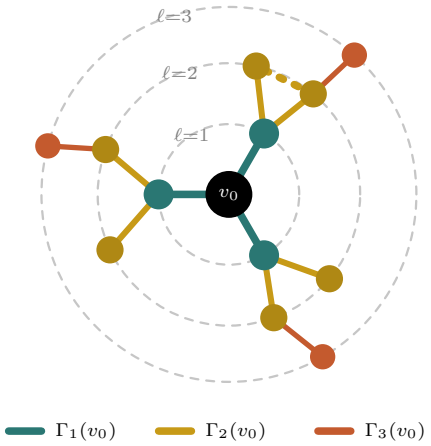
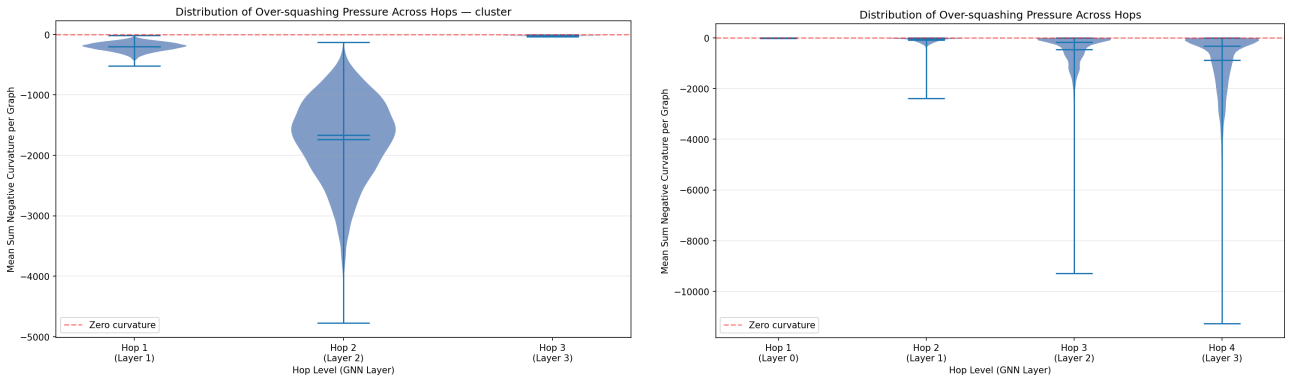


Figure 14. **Hop rings around a source v_0.** Each hop ℓ contributes one non-overlapping ring $\Gamma_\ell(v_0)$, the edges whose farther endpoint is at distance exactly ℓ. Solid lines are radial edges (endpoints one hop apart); the dashed line is a lateral edge (both endpoints at the same hop). $S_{v_0}^{(\ell)}$ sums the negatively curved edges within a ring.

R.4. BFC Profiles on Real Graphs: CLUSTER vs MalNet-Tiny

To illustrate that the hop-stratified BFC pressure $\bar{S}^{(\ell)}$ defined in Section 2.4 captures meaningful structural variation across real graph datasets, Figure 15 compares per-graph distributions of $S_{v_0}^{(\ell)}$ across hops on CLUSTER and MalNet-Tiny. Two levels of variation are visible. First, the dataset-level profiles are qualitatively distinct: CLUSTER concentrates negative-curvature pressure sharply at Hop 2 (median ≈ -1700) and decays to near zero by Hop 3, whereas MalNet-Tiny shows monotonically deepening pressure across hops, reaching extreme values (tail ≈ -10000) at Hop 4. Second, within each dataset, the per-graph distribution at every hop is wide and asymmetric, with long negative tails and substantial across-graph variance. Both observations support the same claim: bottleneck pressure varies non-trivially across depths and across graphs, so the question of whether per-layer GNN choice should respond to this variation remains structurally open. The relationship between these profiles and consensus-architecture per-layer preferences is left to future work.



(a) CLUSTER *(b)* MalNet-Tiny

Figure 15. Hop-stratified over-squashing pressure $S_{v_0}^{(\ell)}$ across hops on (a) CLUSTER and (b) MalNet-Tiny. Each violin shows the distribution of per-graph pressure at a given hop: lower (more negative) values indicate stronger bottleneck concentration. Two levels of variation are visible: (i) the dataset-level profiles differ qualitatively — CLUSTER concentrates pressure at Hop 2 then decays, MalNet-Tiny accumulates progressively across all four hops; (ii) within each dataset, the per-graph distribution is wide and asymmetric at every hop, indicating substantial graph-level heterogeneity. Both observations are consistent, leaving the question of whether the choice of per-layer GNN should respond to depth-varying graph geometry structurally open. Individual graph example for MalNet demonstrated in Figure 1.

S. Operator Implementation Details

This appendix provides the core message-passing equations for the candidate GNN operators in the two search spaces $\mathbf{SS_1} = \{\text{GIN}, \text{GCN}, \text{GAT}, \text{SAGE}\}$ and $\mathbf{SS_2} = \{\text{GINE}, \text{GatedGCN}, \text{GATv2}\}$, as used in our experiments. Throughout, $\mathbf{h}_i^{(\ell)} \in \mathbb{R}^d$ denotes the feature vector of node i at layer ℓ, $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$ denotes the edge feature on edge (i, j), $\mathcal{N}(i)$ denotes the set of neighbours of i, and $\tilde{\mathcal{N}}(i) = \mathcal{N}(i) \cup \{i\}$ includes the self-loop.

Implementation. All operators are used exactly as implemented in the GPS Graph Transformer codebase (Rampášek et al., 2022), with identical hyperparameters where applicable. GIN, GCN, GAT, SAGE, GINE, and GATv2 use the underlying PyTorch Geometric (Fey & Lenssen, 2019) implementations (`GINConv`, `GCNConv`, `GATConv`, `SAGEConv`, `GINEConv`, `GATv2Conv`); GatedGCN uses the custom `GatedGCNLayer` from the GPS repository.

Table 18. **Operators in Search Space 1 (SS1).** GCN and GAT aggregate over the closed neighbourhood $\tilde{\mathcal{N}}(i) = \mathcal{N}(i) \cup \{i\}$, while GIN and SAGE aggregate over $\mathcal{N}(i)$ with the self term added explicitly. $\tilde{d}_i = |\tilde{\mathcal{N}}(i)|$ is degree with self-loop; σ is the nonlinearity (ReLU).

Operator	Update equation
GIN (Xu et al., 2019)	$\mathbf{h}_i^{(\ell+1)} = \text{MLP}\Big((1+\epsilon)\mathbf{h}_i^{(\ell)} + \sum_{j\in\mathcal{N}(i)} \mathbf{h}_j^{(\ell)}\Big)$
GCN (Kipf & Welling, 2017)	$\mathbf{h}_i^{(\ell+1)} = \sigma\Big(\sum_{j\in\tilde{\mathcal{N}}(i)} \frac{1}{\sqrt{\tilde{d}_i\tilde{d}_j}} \mathbf{W}^{(\ell)}\mathbf{h}_j^{(\ell)}\Big)$
GAT (Veličković et al., 2018)	$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j\in\tilde{\mathcal{N}}(i)} \alpha_{ij}^{(\ell)}\mathbf{W}^{(\ell)}\mathbf{h}_j^{(\ell)}\right),$ $\alpha_{ij}^{(\ell)} = \text{softmax}_j\Big(\text{LeakyReLU}\Big(\mathbf{a}^\top[\mathbf{W}^{(\ell)}\mathbf{h}_i^{(\ell)} \,\|\, \mathbf{W}^{(\ell)}\mathbf{h}_j^{(\ell)}]\Big)\Big)$
SAGE (Hamilton et al., 2017)	$\mathbf{h}_i^{(\ell+1)} = \sigma\Big(\mathbf{W}^{(\ell)} \cdot [\mathbf{h}_i^{(\ell)} \,\|\, \max_{j\in\mathcal{N}(i)} \mathbf{h}_j^{(\ell)}]\Big)$

S.1. Search Space 2 (SS2): Edge-Aware Operators

The operators in SS2 incorporate edge features $\mathbf{e}_{ij}$ directly into message construction.

GINE (Hu et al., 2020). The edge-aware extension of GIN. With edge features projected to the node feature dimension via a linear map $\text{Lin}_e : \mathbb{R}^{d_e} \to \mathbb{R}^d$,

$$\mathbf{h}_i^{(\ell+1)} = \text{MLP}\left((1+\epsilon)\mathbf{h}_i^{(\ell)} + \sum_{j\in\mathcal{N}(i)} \text{ReLU}\Big(\mathbf{h}_j^{(\ell)} + \text{Lin}_e(\mathbf{e}_{ij})\Big)\right). \tag{32}$$

GatedGCN (Bresson & Laurent, 2017). GatedGCN maintains both node and edge representations through layers, with gated message propagation. At each layer, edge features $\mathbf{e}_{ij}^{(\ell)}$ are updated alongside node features:

$$\mathbf{e}_{ij}^{(\ell+1)} = \mathbf{D}\mathbf{h}_i^{(\ell)} + \mathbf{E}\mathbf{h}_j^{(\ell)} + \mathbf{C}\mathbf{e}_{ij}^{(\ell)}, \tag{33}$$

$$\boldsymbol{\sigma}_{ij} = \text{sigmoid}\Big(\mathbf{e}_{ij}^{(\ell+1)}\Big), \tag{34}$$

$$\mathbf{h}_i^{(\ell+1)} = \mathbf{A}\mathbf{h}_i^{(\ell)} + \frac{\sum_{j\in\mathcal{N}(i)} \boldsymbol{\sigma}_{ij} \odot \mathbf{B}\mathbf{h}_j^{(\ell)}}{\sum_{j\in\mathcal{N}(i)} \boldsymbol{\sigma}_{ij} + \epsilon}, \tag{35}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$ are layer-specific learnable linear transformations and $\odot$ is element-wise multiplication. Both $\mathbf{h}^{(\ell+1)}$ and $\mathbf{e}^{(\ell+1)}$ pass to the next layer. For clarity, we show the core gated message passing; the `GatedGCNLayer` additionally applies batch normalisation, activation, dropout, and residual connections to both the node and edge streams, following the GraphGPS implementation.

GATv2 (Brody et al., 2022). GATv2 fixes the static-attention problem of the original GAT by applying the attention scoring nonlinearity *before* the projection, yielding dynamic attention. With edge features incorporated into the scoring,

$$\alpha_{ij}^{(\ell)} = \mathrm{softmax}_j\left(\mathbf{a}^\top \mathrm{LeakyReLU}\left(\mathbf{\Theta}_s \mathbf{h}_i^{(\ell)} + \mathbf{\Theta}_t \mathbf{h}_j^{(\ell)} + \mathbf{\Theta}_e \mathbf{e}_{ij}\right)\right), \tag{36}$$

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j\in\tilde{\mathcal{N}}(i)} \alpha_{ij}^{(\ell)}\, \mathbf{\Theta}_t \mathbf{h}_j^{(\ell)}\right). \tag{37}$$

T. Brute-Force Ranking

Figures 16 and 17 show the full brute-force ranking landscape for the five small-enough configurations used to validate DARTS in the main paper (Section 3, Table 2). Each panel corresponds to one (dataset, search space) pair, with all candidate architectures plotted as grey bars sorted by test accuracy or validation accuracy. The DARTS hybrid-consensus architecture is highlighted in blue and the random-ensemble baseline (where run, on SS2 cases) in red, with the rank annotated. The MalNet-Tiny SS2 panel ($k = 81$ architectures) is the most decisive: DARTS lands in the top third while the random ensemble falls to the bottom quartile. On the smaller Peptides spaces ($k = 16$ for SS1, $k = 9$ for SS2), DARTS consistently lands at or near rank 1, while the random ensemble produces vote distributions that fail to reject uniformity (Fisher $p > 0.5$ across all three SS2 cases; full statistics in Table 2).

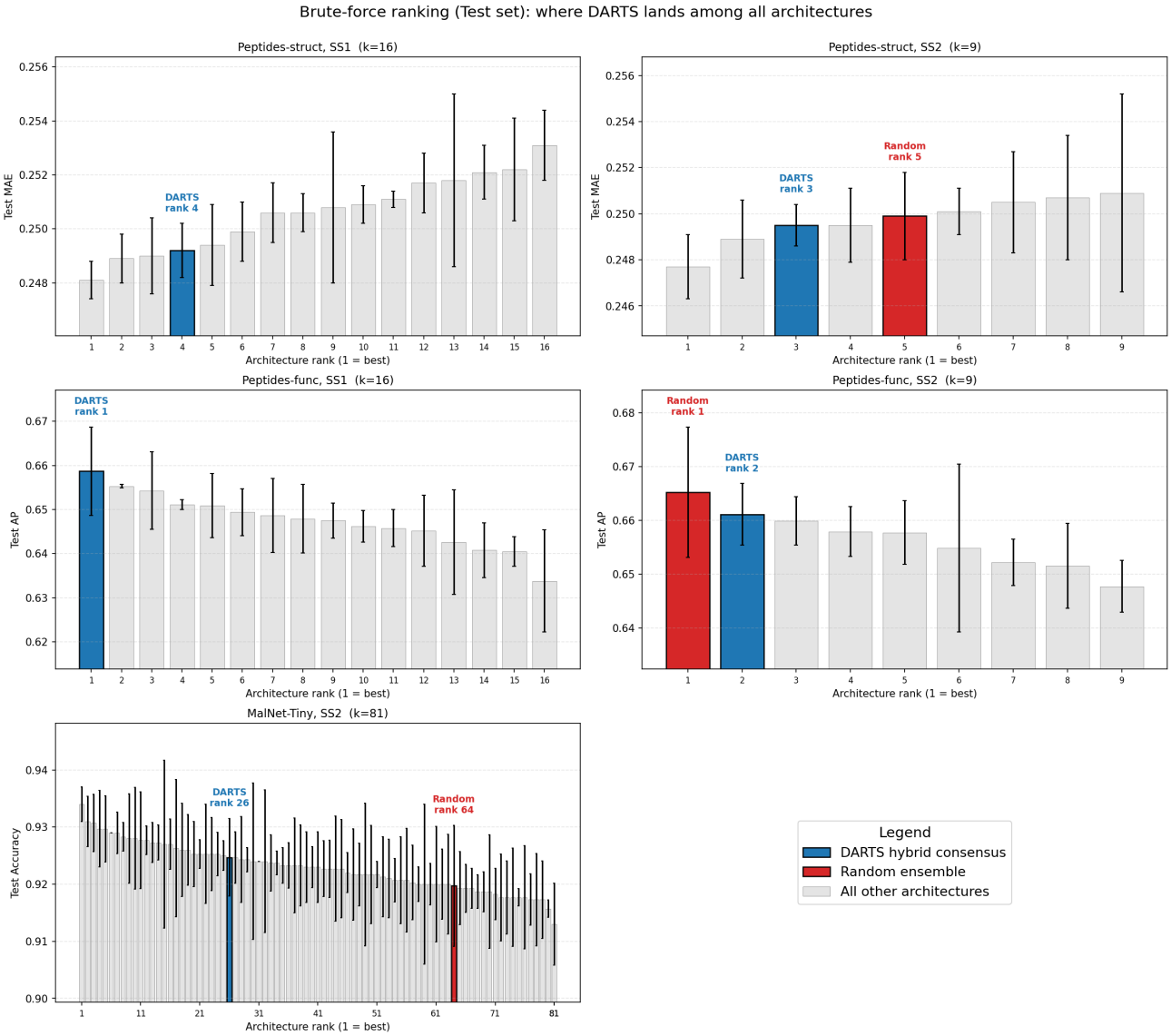


Figure 16. Brute-force test-set ranking for five configurations: Peptides-struct SS1 (k=16), Peptides-struct SS2 (k=9), Peptides-func SS1 (k=16), Peptides-func SS2 (k=9), MalNet-Tiny SS2 (k=81). Blue: DARTS hybrid consensus. Red: random ensemble (SS2 only).

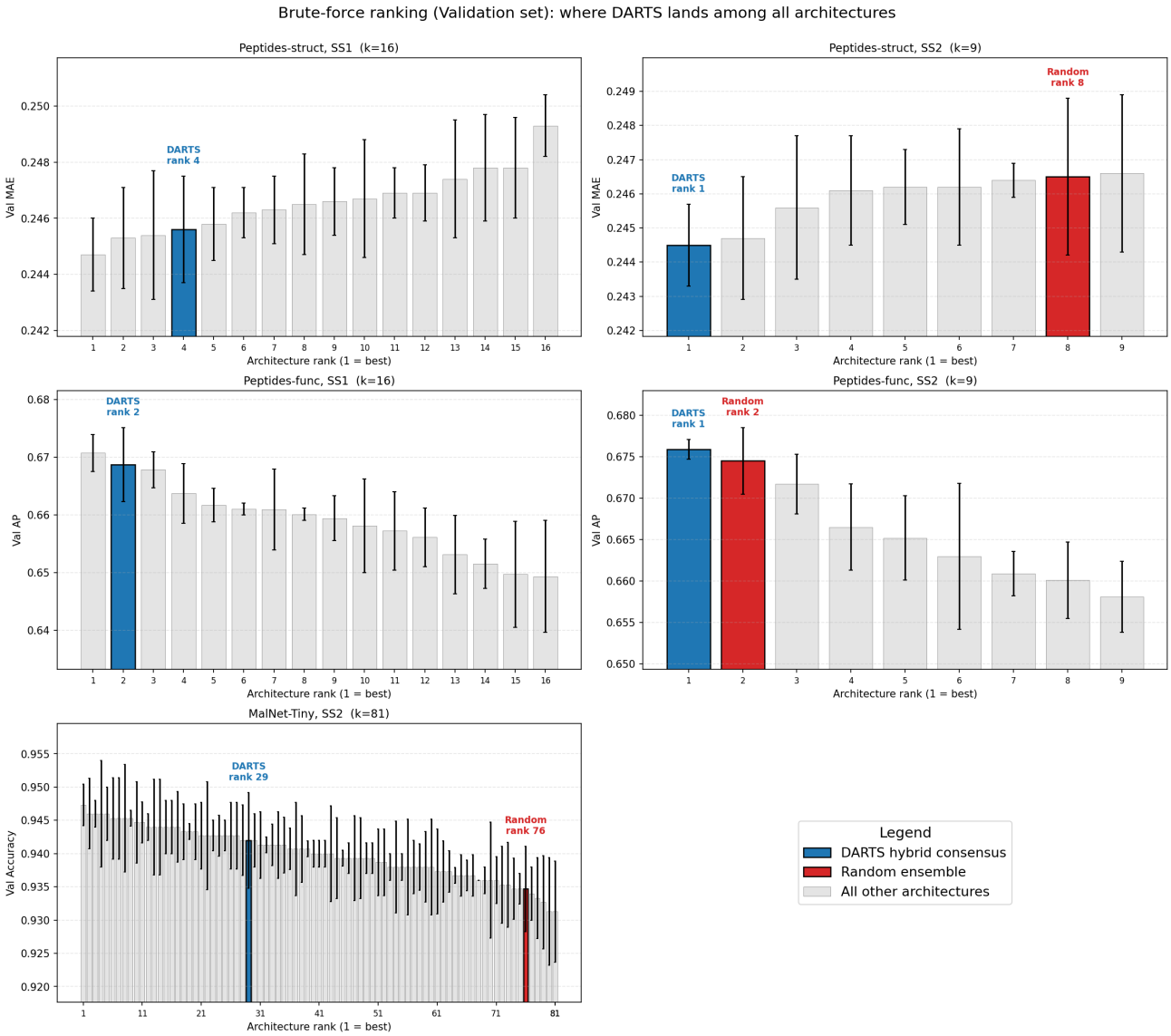
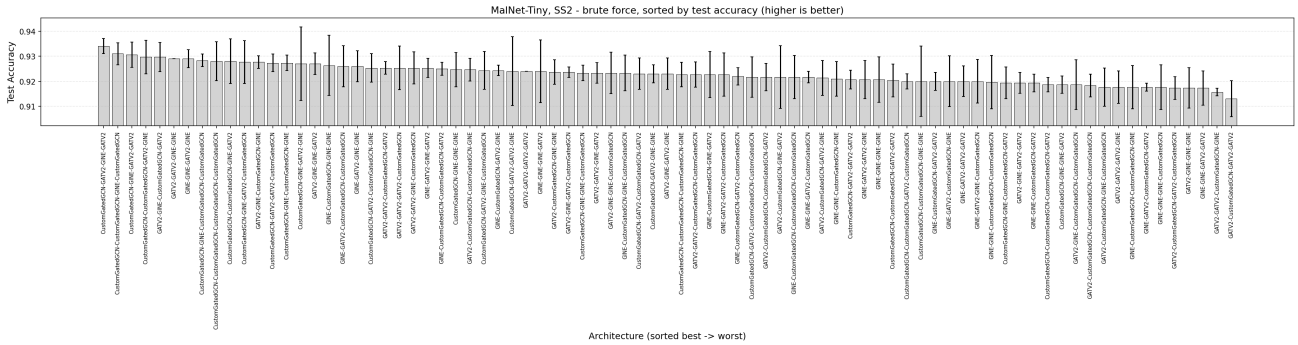
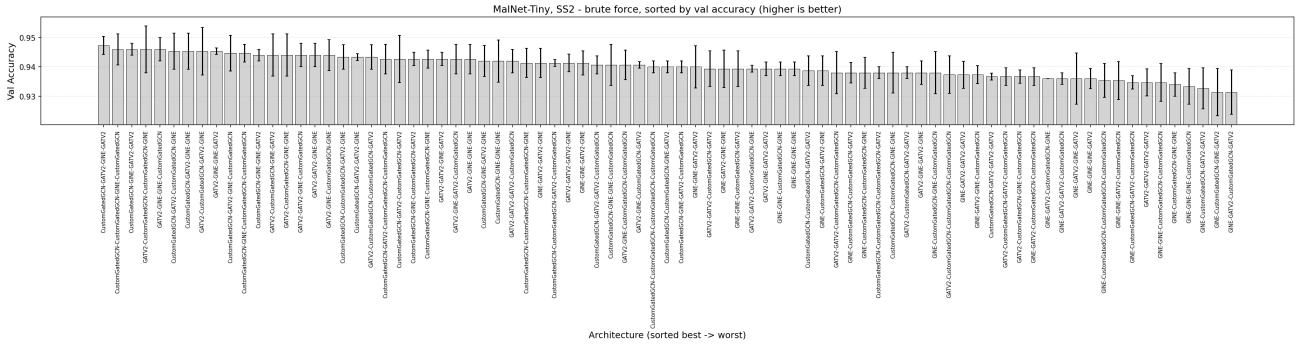


Figure 17. Brute-force validation-set ranking for the same five configurations as Figure 16.



(a) Test set: CGCN^4 ranks 9th.



(b) Validation set: CGCN^4 ranks 40th

Figure 18. Detailed brute-force architecture ranking on MalNet-Tiny SS2. The uniform CGCN^4 architecture (closest to the global minimum) ranks 9 on test but 40 on validation, indicating substantial variability among homogeneous architectures across splits.

U. Ensemble Robustness at $N = 20$ on MalNet-Tiny

To test whether the consensus architecture is sensitive to the choice of $N = 10$ seeds, we additionally ran $N = 20$ seeds on MalNet-Tiny SS2 (Entangled-GT). The hybrid-consensus architecture is identical to the $N = 10$ result: $[\text{CustomGatedGCN}, \text{GINE}, \text{GINE}, \text{GINE}]$. Per-layer vote distributions are CustomGatedGCN $18/20$ at $\ell = 0$, GINE $12/20$ at $\ell = 1$, GINE $13/20$ at $\ell = 2$, and a mean-rule fallback at $\ell = 3$ (GINE $11/20$, below the $0.6N$ threshold). Evidently, we can observe from Figure 18 that the CGCN has been the most popular among the top architectures; hence, DARTS is choosing a reliable architecture. Doubling the seed count does not change the consensus or the methodological pattern, supporting that $N = 10$ is sufficient for stable per-layer preferences in this setting.

V. Layer-wise Heatmaps: SS1 and SS2

Figure 19 contrasts a homogeneous backbone, which repeats one operator at every depth, with a heterogeneous per-layer architecture, where the operator varies by layer; the heatmaps below show this heterogeneity as it actually appears in the learned consensus architectures. Figures 20 and 21 show the per-layer consensus operator across all eight datasets in SS1 and SS2, respectively. Markers indicate hybrid-rule fallbacks at low-confidence layers (where no operator achieved the $0.6N$ vote majority and the mean architecture parameter was used instead). Within each dataset, operator selection varies across layers, and the per-layer pattern differs across datasets, confirming that layer-wise heterogeneity is dataset-specific rather than driven by a global preference for one operator.

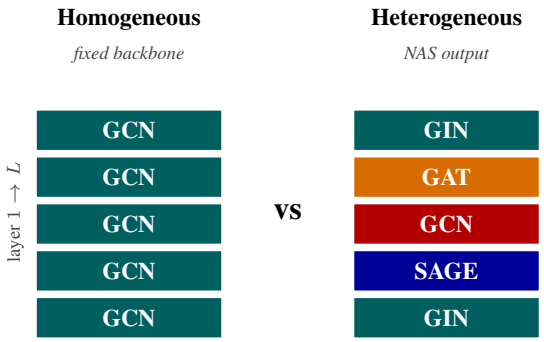


Figure 19. **Homogeneous vs heterogeneous per-layer architectures.** The standard fixed backbone repeats one operator at every depth (left). Neural architecture search instead returns a different operator per layer (right). Each cell is the message-passing operator at that layer; colours denote distinct operators.

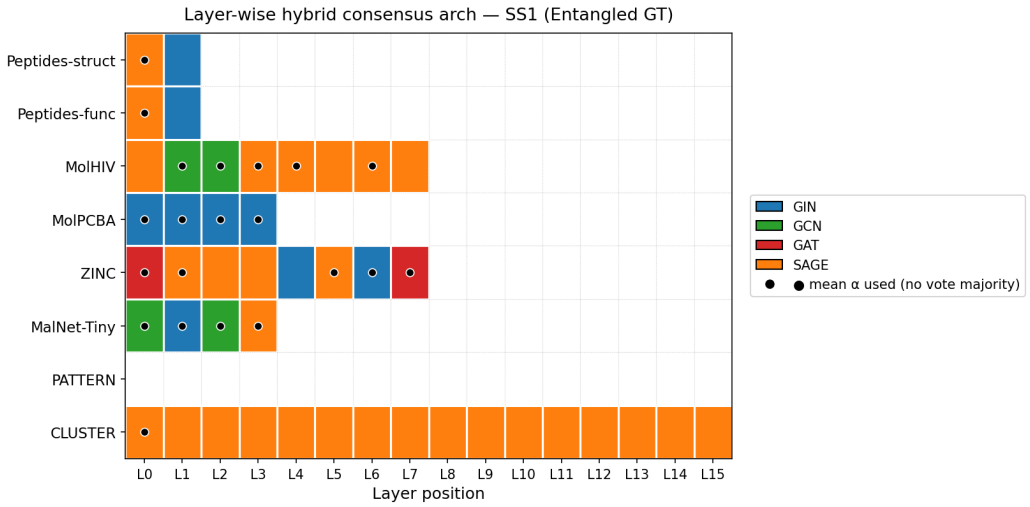


Figure 20. Layer-wise hybrid-consensus operator heatmap on SS1 ({GIN, GCN, GAT, SAGE}) across eight datasets, Entangled-GT backbone.

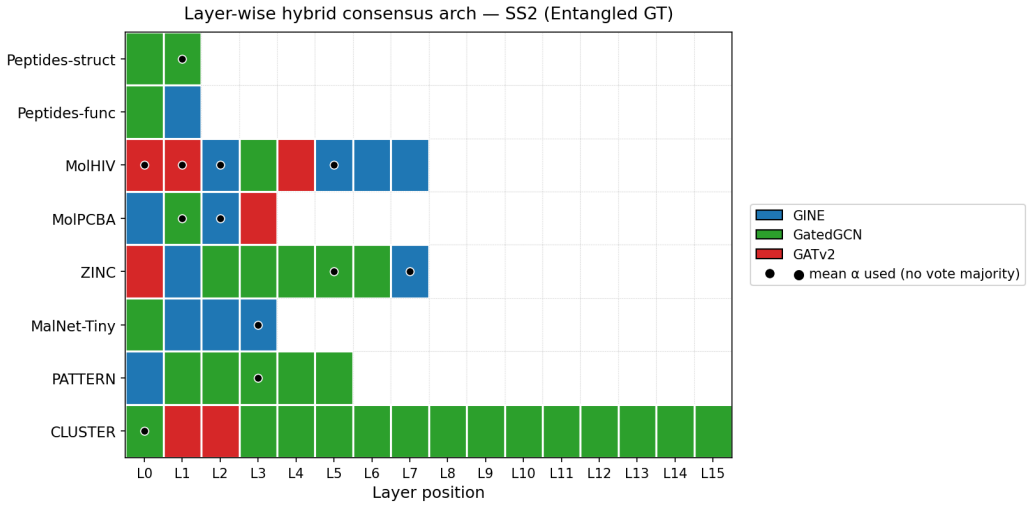


Figure 21. Layer-wise hybrid-consensus operator heatmap on SS2 ({GINE, GatedGCN, GATv2}) across eight datasets, Entangled-GT backbone.

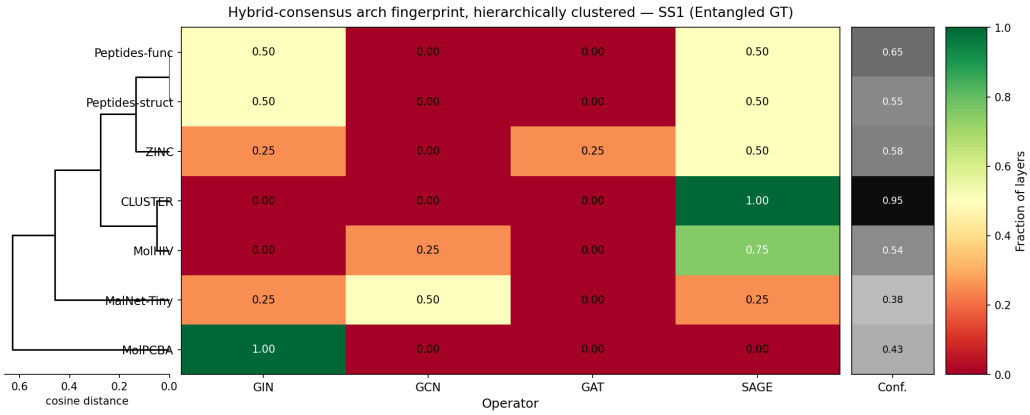


Figure 22. Cluster dendrogram SS1 + per-layer confidence column