QUALITY BEFORE QUANTITY: CURVATURE- AND ATTENTION-ALIGNED SPARSE REWIRING FOR GRAPH NEURAL NETWORKS

Anonymous Authors
Paper under double-blind review

ABSTRACT

Graph rewiring can shorten communication paths in a message-passing graph, but a new edge is useful only if the downstream attention model actually uses it. This issue is especially important for graph attention networks, where added edges compete with existing neighbors for attention: a shortcut may be structurally attractive yet receive little attention. We call this mismatch the *rewiring–utilization gap*. We introduce CARAT (Curvature- and Attention-Aligned Rewiring with Adaptive Topology), a sparse rewiring framework that asks three questions: where should edges be added, which candidate edges are useful, and how strongly should they be used? First, exact Balanced Forman curvature identifies bottlenecked two-hop paths, while a top-k budget per source node prevents uncontrolled graph growth. Second, a learned directional compatibility score estimates whether the endpoints of a candidate shortcut are useful to connect for an attention model. Third, a monotone soft gate lets the task adapt shortcut strength end-to-end while preserving all original edges. Across Cora, Citeseer, filtered Chameleon and Squirrel, and the Peptides-func/Peptides-struct long-range benchmarks, we find that rewiring is strongly regime-dependent: it can hurt graphs that already support effective local propagation, while it is valuable when long-range communication matters. On Peptides-func, CARAT improves average precision from 0.4747 for GATv2 to 0.5412; on Peptides-struct it reduces MAE from 0.4216 to 0.3671. A budget-matched uniform two-hop baseline captures much of this gain, but CARAT gives the best mean on both tasks. We also find that irreversible hard pruning is not consistently helpful. These results suggest a practical principle: for attention GNNs, shortcut quality and soft task adaptation matter more than simply adding more edges.

1 INTRODUCTION

Message-passing graph neural networks (GNNs) learn by repeatedly exchanging information along graph edges (Gilmer et al., 2017; Bronstein et al., 2021). This local computation is attractive because it is sparse and scalable, but it also creates a basic limitation: information that matters several hops away must pass through every intermediate neighborhood. When many distant signals are compressed through a narrow part of the graph, their influence can become weak, a phenomenon commonly called *over-squashing* (Alon and Yahav, 2021; Topping et al., 2022; Di Giovanni et al., 2023; Black et al., 2023). A natural response is graph rewiring: add a small number of shortcut edges so that information does not need to travel through the original bottleneck.

The idea sounds simple, but it hides a question that becomes important for attention-based GNNs. *If a rewiring method adds an edge, will the attention model actually use it?* Structural rewiring and learned attention solve two different problems. Rewiring changes where information *can* flow. Attention decides how much information *does* flow. Prior work has used curvature, effective resistance, spectral objectives, feature similarity, communities, and joint graph–feature denoising to guide graph modification (Topping et al., 2022; Black et al., 2023; Arnaiz-Rodríguez et al., 2022; Karhadkar et al., 2023; Jamadandi et al., 2024; Rubio-Madrigal et al., 2025; Linkerhägner et al., 2025). Other work has already explored combining rewiring and graph attention, although with a different structural mechanism (Liao and Póczos, 2025). Recent evaluations also caution that

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

curvature-rewiring gains can be highly sensitive to hyperparameters and experimental protocol (Tori et al., 2025). These observations motivate us to treat shortcut creation and shortcut utilization as separate objects, with particular focus on whether structurally proposed shortcuts are actually used by the attention mechanism.

We call their mismatch the **rewiring–utilization gap**. Consider an edge (i, j) added by a structural rule. In a GAT-style layer, its normalized contribution is determined by a softmax over the competing neighbors of i (Veličković et al., 2018; Brody et al., 2022). A shortcut may therefore exist in the computational graph but have almost no functional effect. Conversely, indiscriminately adding many two-hop edges can dilute attention, increase memory cost, and damage graphs whose local structure was already useful.

This paper studies that gap through CARAT, short for *Curvature- and Attention-Aligned Rewiring with Adaptive Topology*. CARAT uses exact Balanced Forman curvature (BFC) to score bottleneck severity, learns a directional compatibility function for candidate endpoints, and keeps the candidate set bounded by selecting at most k two-hop shortcuts per source node. Each shortcut receives a positive, monotone soft gate that is optimized with the downstream task. Original edges are never removed by the main method. We also study a hard consolidation variant, CARAT-H, in which low-utilization shortcuts may be removed after warm-up.

Our experiments lead to a more nuanced conclusion than "rewiring always helps." On Cora, every tested rewiring variant is worse than plain GATv2. On Citeseer, controlled rewiring gives a modest improvement. On filtered Chameleon and Squirrel (Platonov et al., 2023), the effect is small, but quality control avoids the larger degradation caused by uniform shortcuts on Squirrel. The clearest gains appear on the Long Range Graph Benchmark (LRGB) (Dwivedi et al., 2022). On Peptides-func, GATv2 improves from 0.4747 AP to 0.5380 simply by receiving a budget-matched set of two-hop shortcuts, while CARAT reaches 0.5412. On Peptides-struct, MAE drops from 0.4216 to 0.3694 with uniform shortcuts and to 0.3671 with CARAT. These results show that access to shorter communication paths is itself a strong effect, while careful shortcut adaptation gives a smaller additional benefit. Hard pruning, in contrast, is not consistently useful.

Our contributions are therefore deliberately focused:

- We formalize the *rewiring–utilization gap*: structural availability of a shortcut does not imply functional use by an attention GNN.
- We propose CARAT, a bounded sparse rewiring mechanism that combines exact curvature-guided candidate discovery, learned directional compatibility, and differentiable shortcut gating while preserving the original graph.
- We provide controlled experiments that separate the effect of *having short paths* from the effect of *selecting and adapting shortcuts*. The results show strong regime dependence and expose when hard structural decisions are unnecessary or harmful.

2 BACKGROUND AND RELATED WORK

Message passing and graph attention. GCNs, GIN-style models, and related message-passing architectures aggregate information from local neighborhoods (Kipf and Welling, 2017; Gilmer et al., 2017; Xu et al., 2019). GAT replaces fixed aggregation weights with learned attention coefficients (Veličković et al., 2018); GATv2 changes the attention formulation so that neighbor ranking can depend on the query node (Brody et al., 2022). Our work uses GATv2 as a controlled backbone because the central question is not whether attention is useful, but whether a structurally added edge is functionally utilized after attention normalization.

Over-squashing and rewiring. Over-squashing describes weak sensitivity to distant information when many signals must pass through limited graph connectivity (Alon and Yahav, 2021). Curvature gives one structural view: Topping et al. (2022) connect negatively curved edges to bottlenecks and introduce curvature-driven rewiring. Complementary analyses emphasize commute time and effective resistance (Di Giovanni et al., 2023; Black et al., 2023). Rewiring methods then differ in the structural quantity they optimize. DiffWire learns commute-time/spectral reweighting (Arnaiz-Rodríguez et al., 2022), FoSR adds edges using first-order spectral expansion (Karhadkar et al., 2023), and SJLR explicitly studies the trade-off between over-squashing and over-smoothing by adding and deleting

edges (Giraldo et al., 2023). Spectral graph pruning further shows that edge deletion can sometimes improve both information flow and smoothing behavior (Jamadandi et al., 2024). These results make an important point for our study: improving structural connectivity does not necessarily improve downstream task performance, so more connectivity is not automatically better.

Feature-aware, differentiable, and attention-coupled rewiring. Beyond purely structural rewiring, recent work incorporates features, differentiability, or attention. ComFy uses community structure and feature similarity to guide graph modification (Rubio-Madrigal et al., 2025), while JDR jointly rewires the graph and denoises features through spectral alignment (Linkerhägner et al., 2025). Differentiable approaches such as DiffWire optimize a differentiable graph objective (Arnaiz-Rodríguez et al., 2022), while Greener GRASS couples structural encoding, stochastic rewiring, and graph attention in one architecture (Liao and Póczos, 2025). We therefore do not claim that feature-aware rewiring or rewiring-plus-attention is new by itself. Our narrower focus is the interface between *a proposed shortcut and its learned use*: CARAT keeps candidate growth bounded, learns a directional compatibility score, and exposes each shortcut through a soft structural gate to a GATv2 backbone.

Heterophily. Heterophilous graphs are a difficult regime for local neighborhood aggregation, and prior work has proposed both architectural designs such as H2GCN and strong non-homophilous baselines such as LINKX (Zhu et al., 2020; Lim et al., 2021).

Long-range benchmarks and evaluation caution. LRGB was designed to stress long-range interaction in graph learning and includes Peptides-func and Peptides-struct (Dwivedi et al., 2022). Global-attention models such as GraphGPS provide a different route to long-range communication (Rampášek et al., 2022); our study instead asks how far a sparse GATv2 backbone can be improved by controlled shortcuts. Finally, graph rewiring is sensitive to graph version, edge budget, and model hyperparameters. Tori et al. (2025) show that conclusions about curvature rewiring can change substantially under careful tuning. We therefore budget-match the uniform two-hop control, select the node-level candidate budget using validation only, and report multiple seeds.

Edge removal. Random edge removal has a separate history as regularization, for example DropEdge (Rong et al., 2020). Our hard variant differs because it removes only added shortcuts after warm-up, based on learned utilization, and never deletes original graph edges. The main CARAT model remains soft and does not require irreversible pruning.

3 THE REWIRING–UTILIZATION GAP

Consider a two-hop pair (i, j) identified as a useful structural shortcut. Once added, it competes with existing neighbors for attention, but its structural importance does not determine how much attention it receives. Let e_{ij} be the pre-softmax attention score assigned by an attention head to edge (i, j) and let

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}. \tag{1}$$

Suppose a rewiring rule adds (i, j) because it improves a structural criterion. At this point, the edge is only structurally available; whether it contributes to message passing is determined later by the attention mechanism. Let $e_{i\star} = \max_{k \in \mathcal{N}(i)} e_{ik}$. If $e_{ij} \leq e_{i\star} - \Delta$ for $\Delta \geq 0$, then

$$\alpha_{ij} \leq \frac{\exp(e_{i\star} - \Delta)}{\exp(e_{i\star})} = \exp(-\Delta). \tag{2}$$

Thus a structurally desirable edge can have exponentially small normalized attention relative to a better-matched neighbor. Structural criteria therefore determine which edges are available to the model, but do not by themselves determine which edges will contribute strongly to message passing. Equation (2) is intentionally simple; its role is not to prove that a particular curvature score determines attention, but to make the design problem explicit. **Edge existence and edge utilization are different variables.**

This gap motivates three design constraints. First, the candidate pool should be bounded so that added shortcuts do not overwhelm the attention neighborhood; CARAT enforces this with top-k selection.

Second, candidate shortcuts should be screened for task-relevant compatibility; CARAT uses a learned directional compatibility score for this purpose. Third, shortcut strength should remain adaptive to the downstream task rather than being fixed by a preprocessing heuristic; CARAT uses a monotone soft gate to provide this adaptation.

4 CARAT: CURVATURE- AND ATTENTION-ALIGNED REWIRING

Figure 1 gives the full pipeline. CARAT is designed to address the rewiring–utilization gap by separating shortcut creation from shortcut utilization. The main model preserves all original edges, proposes a bounded set of two-hop shortcuts, learns their compatibility with the attention model, and uses a trainable soft gate to expose their strength to GATv2 as a structural edge attribute. A hard consolidation step is studied only as an optional variant.

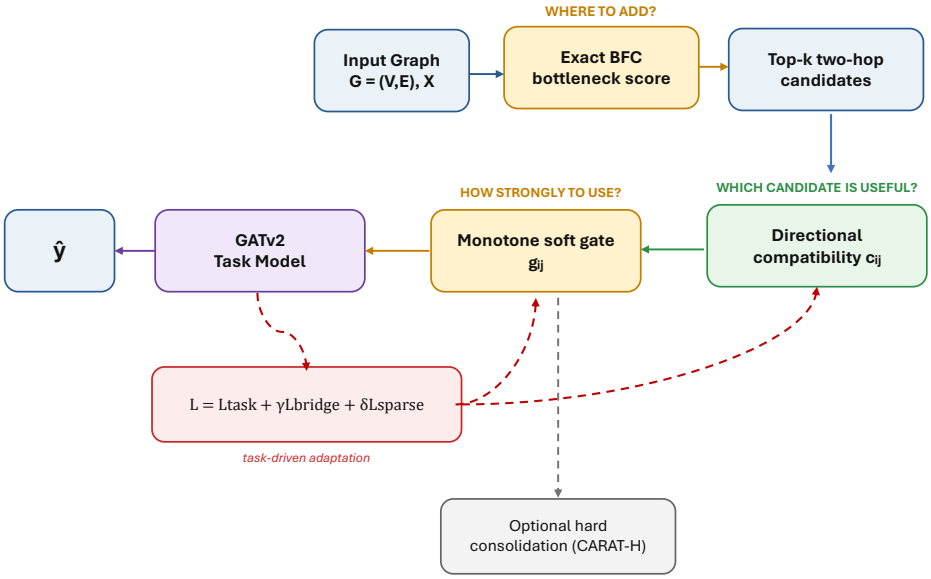


Figure 1: **CARAT overview.** Curvature narrows the structural search space; learned directional compatibility and a soft gate adapt shortcuts to the attention model. The original graph is always retained. Hard consolidation is optional and is not part of the canonical CARAT model.

4.1 CURVATURE-GUIDED BOUNDED CANDIDATE DISCOVERY

We start from exact Balanced Forman curvature (Topping et al., 2022). For each original edge (u, v), we compute its BFC value $\kappa_{uv}^{\mathrm{BFC}}$, including the triangle and non-triangulated four-cycle correction used in the original definition. We convert negative curvature into a bounded bottleneck severity

$$b_{uv} = \frac{\mathrm{clip}(-\kappa_{uv}^{\mathrm{BFC}}, 0, 2)}{2} \in [0, 1]. \tag{3}$$

The exact code-matched BFC expression is given in Appendix A.

A candidate shortcut (u, v) must connect nodes at distance two in the current graph and must not already be an edge. Let

$$\mathcal{P}_{uv} = \{m : (u, m) \in E,\ (m, v) \in E\} \tag{4}$$

be the set of middle nodes on two-hop paths. We score a candidate by the curvature severity of those paths:

$$t_{uv} = \mathrm{clip}\left(\frac{\sum_{m \in \mathcal{P}_{uv}} (b_{um} + b_{mv})}{2|\mathcal{P}_{uv}|^{3/2}}, 0, 1\right). \tag{5}$$

The $|\mathcal{P}_{uv}|^{-1/2}$ factor favors shortcuts across narrow regions over pairs that already have many redundant two-hop routes. For each source node, we retain at most the top k candidates by t_{uv}.

Therefore the directed candidate budget is at most $k|V|$ before optional symmetrization. This budget is the main safeguard against the candidate explosion that occurs when all two-hop pairs are inserted.

4.2 Attention-aligned directional compatibility

Raw cosine similarity is a weak proxy for attention, especially on heterophilic graphs where useful communication may occur between dissimilar raw features. To determine which candidate shortcuts are useful to the attention model, we therefore learn a directional compatibility score. For node feature x_i, define

$$z_i = \mathrm{ELU}(W_0 x_i), \qquad q_i = \frac{W_q z_i}{\|W_q z_i\|_2}, \qquad k_i = \frac{W_k z_i}{\|W_k z_i\|_2}, \tag{6}$$

then use the directional score

$$c_{ij} = q_i^\top k_j \in [-1, 1]. \tag{7}$$

Because q_i and k_j use different projections, c_{ij} need not equal c_{ji}. This directionality reflects the asymmetric roles of a source node and its neighbor in attention: the relevance of node j as a message source for node i need not equal that of node i for node j.

4.3 Monotone soft shortcut gating

For candidate (i, j), topology captures structural value while compatibility captures fit to the attention model; CARAT combines them into a score that determines adaptive shortcut strength:

$$s_{ij} = \lambda t_{ij} + (1 - \lambda) c_{ij}, \quad \lambda = \sigma(\lambda_{\mathrm{raw}}). \tag{8}$$

The gate is

$$g_{ij} = \sigma(\alpha s_{ij} + \beta), \qquad \alpha = \mathrm{softplus}(\alpha_{\mathrm{raw}}) + \epsilon > 0. \tag{9}$$

The gate provides a differentiable way for the downstream task to determine how strongly a structurally proposed shortcut should be exposed to the attention model. The positive parameterization of α makes the gate monotone in the intended score: stronger bottleneck evidence or stronger learned compatibility cannot flip direction because of a negative temperature. Original edges have gate 1 and are always preserved.

We pass $[g_{ij}, \mathbb{1}\{(i, j) \in E\}]$ as structural edge attributes to GATv2. The gate is therefore not a hard multiplicative mask on the message. It is a learned structural signal that the attention layer can use together with node representations.

4.4 Shortcut-only structural regularization

The task loss alone may not ensure that candidate shortcuts remain useful, while activating too many can undermine sparsity. We therefore introduce two regularization terms on candidate shortcuts: a *bridge loss* that preserves structurally important shortcuts and a *sparsity loss* that discourages unnecessary activation:

$$\mathcal{L}_{\mathrm{bridge}} = -\frac{1}{|\mathcal{E}_{\mathrm{c}}|} \sum_{(i,j) \in \mathcal{E}_{\mathrm{c}}} t_{ij} \log(g_{ij} + \epsilon). \tag{10}$$

The bridge loss assigns a stronger penalty to closing the gate of candidates with high topological scores, encouraging structurally important shortcuts to remain usable.

$$\mathcal{L}_{\mathrm{sparse}} = \frac{1}{|\mathcal{E}_{\mathrm{c}}|} \sum_{(i,j) \in \mathcal{E}_{\mathrm{c}}} g_{ij}. \tag{11}$$

The sparsity loss penalizes the average gate value across candidate shortcuts, discouraging unnecessary activation. The total objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \gamma \mathcal{L}_{\mathrm{bridge}} + \delta \mathcal{L}_{\mathrm{sparse}}. \tag{12}$$

The first term gives high-bottleneck candidates a reason to remain usable; the second charges every shortcut for remaining active. Unlike a global regularizer over the original graph, this objective does not force native edges to compete with speculative shortcuts.

4.5 Optional hard consolidation

We also evaluate CARAT-H, which removes persistently low-utilization shortcuts after a warm-up period. Let $\bar{\alpha}_{ij}$ be attention averaged across heads/layers and let d_j be the destination in-degree. We use a relative attention score $r_{ij} = d_j \bar{\alpha}_{ij}$, for which uniform attention is near one, and define utilization $U_{ij} = g_{ij} r_{ij}$. An EMA of U_{ij} is tracked during warm-up, and a robust median/MAD rule removes only low-utilization shortcuts, never original edges. Appendix C gives the exact rule. The experiments show that hard consolidation is task-dependent; consequently CARAT (soft) is our default method.

5 Experimental Setup

Datasets. For node classification we use Cora and Citeseer as standard homophilous benchmarks, plus the filtered Chameleon-F and Squirrel-F datasets introduced by Platonov et al. (2023) to test rewiring on challenging non-homophilous graphs. The filtered datasets remove duplicate-node leakage present in older WikipediaNetwork versions. We use those filtered versions and avoid claiming a heterophily state of the art. Our filtered graph files contain 890 nodes/$8{,}854$ undirected edges for Chameleon-F and $2{,}223$ nodes/$46{,}998$ undirected edges for Squirrel-F; the PyG message-passing representation stores both directions. We use the provided splits. For long-range graph prediction we use Peptides-func (10-label graph classification, Average Precision) and Peptides-struct (11-target regression, MAE) from LRGB (Dwivedi et al., 2022), which stress long-range graph interactions.

Backbone and variants. All methods use GATv2 (Brody et al., 2022) with matched optimization settings. We compare five variants to isolate the effects of shortcut availability, topology-based selection, learned directional compatibility, and hard consolidation. **(i) GATv2** uses the original graph and serves as the no-rewiring baseline. **(ii) Uniform-2Hop** adds a budget-matched random set of two-hop shortcuts with unit structural edge strength, isolating the benefit of simply providing shorter paths. **(iii) Topology-only** selects shortcuts using BFC-based topology scores without learned directional compatibility and follows the hard consolidation schedule, testing whether topology alone is sufficient for shortcut selection. **(iv) CARAT** is the full soft model, combining BFC-based candidate selection, learned directional compatibility, and task-adaptive soft gating. **(v) CARAT-H** extends CARAT with hard consolidation of persistently low-utilization shortcuts, allowing us to compare soft adaptive retention with irreversible pruning.

Table 1: Comparison of CARAT variants and the components they isolate.

Variant	Shortcuts	BFC	Compatibility	Soft Gate	Hard Consolidation	Purpose
GATv2	No	No	No	No	No	No rewiring
Uniform-2Hop	Yes	No	No	No	No	Effect of short paths
Topology-only	Yes	Yes	No	No	Yes	Effect of BFC selection
CARAT	Yes	Yes	Yes	Yes	No	Full method
CARAT-H	Yes	Yes	Yes	Yes	Yes	Effect of hard consolidation

Protocol. Node experiments use 10 seeds. A single global node candidate budget is selected from $k \in \{3, 5, 8\}$ using validation accuracy of the structurally richest variant (CARAT-H) on Chameleon-F and Squirrel-F only; $k = 8$ is then frozen and shared by every final node method before test evaluation. No test metric is used for this selection. LRGB uses $k = 3$, fixed before the final runs, and five seeds. Node models use hidden size 64, four heads, dropout 0.6, Adam with learning rate 5×10^{-3} and weight decay 5×10^{-4}. LRGB models use four GATv2 layers, hidden size 128, four heads, dropout 0.2, learning rate 10^{-3}, and batch sizes 32/64 for train/evaluation. Full hyperparameters and checkpoint rules are in Appendix E.

6 RESULTS

6.1 NODE CLASSIFICATION: REWIRING IS REGIME-DEPENDENT

Table 2 and Figure 2 show a clear message: the value of rewiring depends on the graph. On Cora, the original graph is best and all shortcut variants lose accuracy. Citeseer behaves differently: every controlled shortcut variant improves over GATv2, with topology-only and CARAT around $+1.4$ percentage points. Chameleon-F is effectively neutral within run-to-run variation. On Squirrel-F, uniform two-hop expansion is harmful (34.13% vs. 36.10% for GATv2), while topology-aware/learned variants recover the loss; CARAT reaches 36.63% and CARAT-H 36.66%.

Table 2: Node classification test accuracy (%, mean ± std., 10 runs). Best mean per dataset is bold. The topology-only variant removes learned compatibility but keeps the hard consolidation schedule.

Method	Cora	Citeseer	Chameleon-F	Squirrel-F
GATv2	**78.98±0.89**	65.99±0.71	40.87±1.63	36.10±2.29
Uniform-2Hop	76.71±0.83	67.03±0.89	40.14±2.90	34.13±1.90
Topology-only	74.85±1.11	**67.37±0.90**	**41.09±3.54**	36.34±1.84
CARAT	74.93±0.93	67.36±0.68	39.77±2.89	36.63±2.55
CARAT-H	74.57±1.14	67.19±1.39	40.02±4.11	**36.66±3.04**

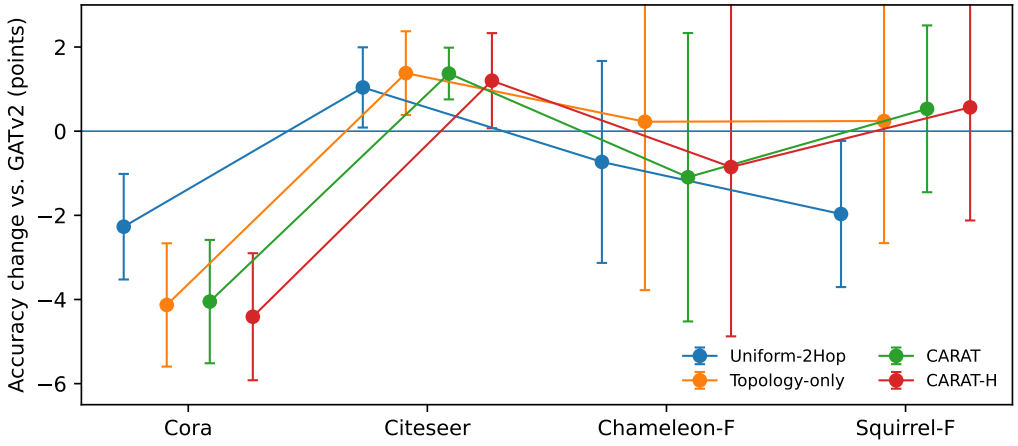


Figure 2: Paired accuracy change relative to GATv2. Error bars show the standard deviation of paired seed differences. Rewiring is not a universally positive intervention: it hurts Cora, helps Citeseer, is neutral on Chameleon-F, and needs quality control to avoid degradation on Squirrel-F.

This result is important for how we interpret graph rewiring. A method that always adds shortcuts can change a graph that did not need to be changed. In our experiments, Cora is the strongest example. The practical goal is therefore not to maximize the number of added edges but to make shortcut creation selective and easy for the task to ignore when unnecessary.

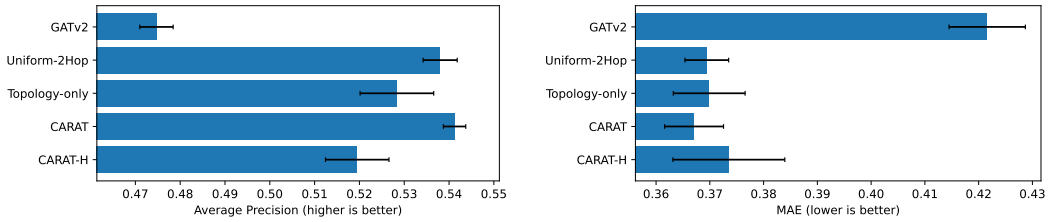
6.2 LONG-RANGE GRAPH TASKS: SHORT PATHS MATTER

The strongest effect appears on the two LRGB peptide tasks (Table 3). On Peptides-func, plain GATv2 obtains 0.4747 AP. Uniform two-hop edges raise this to 0.5380, showing that simply making two-hop communication directly available is already powerful. CARAT gives the best mean, 0.5412. The paired gain of CARAT over Uniform-2Hop is small ($+0.0032$ AP over five seeds) and should not be over-interpreted; a paired test is reported in Appendix H. The larger and more robust conclusion is that shortening communication paths substantially improves this long-range task.

Peptides-struct shows the same pattern. Plain GATv2 has 0.4216 MAE, Uniform-2Hop reduces it to 0.3694, and CARAT gives the best mean at 0.3671. The difference between CARAT and Uniform-2Hop is not statistically clear at five seeds. Thus curvature/compatibility quality control provides a refinement, while access to short paths explains most of the gain.

Table 3: LRGB test results (mean ± std., 5 seeds). Peptides-func uses Average Precision (AP, higher is better); Peptides-struct uses MAE (lower is better).

Method	Peptides-func AP ↑	Peptides-struct MAE ↓
GATv2	0.4747±0.0037	0.4216±0.0071
Uniform-2Hop	0.5380±0.0038	0.3694±0.0041
Topology-only	0.5284±0.0082	0.3699±0.0067
CARAT	**0.5412±0.0025**	**0.3671±0.0055**
CARAT-H	0.5195±0.0071	0.3735±0.0104



(a) Peptides-func

(b) Peptides-struct

Figure 3: Long-range graph results. The dominant effect is the availability of short communication paths; CARAT provides the best mean in both tasks, while hard consolidation reduces performance.

6.3 What do the ablations tell us?

Three observations are consistent across the study. **First, quantity alone is not enough.** Uniform shortcuts help strongly on Peptides but hurt Cora and Squirrel-F. **Second, topology is useful but not sufficient.** Topology-only rewiring is competitive on Citeseer and Chameleon-F, yet on Peptides-func it trails both Uniform-2Hop and CARAT. **Third, hard pruning is not a free improvement.** CARAT-H is close to CARAT on Squirrel-F but worse on both Peptides tasks. Once a shortcut receives a soft gate and is exposed to learned attention, an irreversible deletion can remove an edge that becomes useful later in training.

For this reason, we treat CARAT as the canonical model and hard consolidation as an optional mechanism rather than a required component. This is also why the title emphasizes *quality before quantity*, not aggressive sparsification.

7 Discussion and Limitations

Our results support a practical view of rewiring rather than a universal one. Rewiring is most useful when the task benefits from shorter communication paths. LRGB makes this visible: all controlled two-hop variants are much stronger than the original GATv2. In contrast, Cora shows that adding edges can damage a graph whose local structure is already effective. A method should therefore be able to keep candidate growth bounded and to attenuate shortcuts rather than forcing every proposed edge to have the same role.

The experiments also separate two effects that are sometimes mixed together: the *path-length effect* of making two-hop interactions directly available and the *selection effect* of choosing which shortcuts to emphasize. On Peptides, the path-length effect dominates, while CARAT provides a smaller additional benefit over Uniform-2Hop.

There are several limitations. We use GATv2 as the main backbone and do not claim that the same behavior must hold for every message-passing architecture. Exact BFC requires local motif information and is more expensive than a simple degree-based proxy, although it is computed as structural preprocessing rather than at every message-passing step. Our node experiments are controlled backbone comparisons rather than a state-of-the-art leaderboard; recent methods such as ComFy and JDR solve related but broader structure-learning problems with different pipelines (Rubio-Madrigal et al., 2025; Linkerhägner et al., 2025). Finally, the rewiring–utilization inequality in Section 3 explains why attention can suppress a shortcut, but it is not a theorem that curvature predicts task relevance. Learning that relation remains an open problem.

8 Conclusion

We studied a simple question: after a graph is rewired, does an attention GNN actually use the new edges? CARAT addresses this rewiring–utilization gap with a bounded curvature-guided candidate set, learned directional compatibility, and soft task-adaptive shortcut gates. The experiments show that graph rewiring is not universally beneficial. It can hurt when the original graph already supports the task, but it is highly effective on the long-range Peptides benchmarks. Most of the long-range gain comes from shorter communication paths themselves; CARAT adds a smaller quality-control benefit and gives the best mean on both Peptides tasks. Hard pruning is not consistently helpful. We therefore argue for a conservative principle: add few shortcuts, preserve the original graph, and let the task decide how strongly the new paths should be used.

AI Use Statement

Generative AI tools, including OpenAI ChatGPT and Anthropic Claude, were used for language editing.

Ethics Statement

This work uses public graph-learning benchmark datasets and does not involve human-subject experiments or collection of sensitive personal data. The main research risk is methodological over-claiming; we therefore report negative and neutral results alongside positive ones and avoid claims of universal improvement or state-of-the-art performance.

Reproducibility Statement

The submission package includes the complete anonymized implementation used for all reported experiments, exact BFC computation, deterministic candidate-generation routines, seed-level JSON/CSV outputs, validation-only top-k selection records, and a run manifest containing configuration and source hashes. Appendix E lists dataset processing, hyperparameters, early-stopping rules, and the correspondence between paper names and implementation variants. No test metric was used to select the node candidate budget.

References

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria M. Oliver. DiffWire: Inductive graph rewiring via the lovász bound. In *Proceedings of the First Learning on Graphs Conference*, volume 198 of *Proceedings of Machine Learning Research*, pages 15:1–15:27. PMLR, 2022. URL https://proceedings.mlr.press/v198/arnaiz-rodri-guez22a.html.

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

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

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

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M. Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 7865–7885. PMLR, 2023. URL https://proceedings.mlr.press/v202/di-giovanni23a.html.

Vijay Prakash Dwivedi, Ladislav Rampášek, Mikhail Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, pages 22326–22340, 2022. doi: 10.52202/068431-1622. URL https://papers.neurips.cc/paper_files/paper/2022/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 2017. URL https://proceedings.mlr.press/v70/gilmer17a.html.

Jhony H. Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D. Malliaros. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 566–576. ACM, 2023. doi: 10.1145/3583780.3614997. URL https://doi.org/10.1145/3583780.3614997.

Adarsh Jamadandi, Celia Rubio-Madrigal, and Rebekka Burkholz. Spectral graph pruning against over-squashing and over-smoothing. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0331. URL https://papers.neurips.cc/paper_files/paper/2024/hash/140aac600566125915df7e74ff538f66-Abstract-Conference.html.

Kedar Karhadkar, Pradeep Kr. Banerjee, and Guido Montúfar. FoSR: First-order spectral rewiring for addressing oversquashing in GNNs. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3YjQfCLdrzz.

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

Tongzhou Liao and Barnabás Póczos. Greener GRASS: Enhancing GNNs with encoding, rewiring, and attention. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/647af5f6b2538524f6c047c1d9170fd9-Abstract-Conference.html.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ae816a80e4c1c56caa2eb4e1819cbb2f-Abstract.html.

Jonas Linkerhägner, Cheng Shi, and Ivan Dokmanić. Joint graph rewiring and feature denoising via spectral resonance. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f463d31ed2fdd7b0ec585c041ec1baa8-Abstract-Conference.html.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pages 14501–14515, 2022. doi: 10.52202/068431-1054. URL https://papers.neurips.cc/paper_files/paper/2022/hash/5d4834a159f1547b267a05a4e2b7cf5e-Abstract-Conference.html.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkx1qkrKPr.

Celia Rubio-Madrigal, Adarsh Jamadandi, and Rebekka Burkholz. GNNs getting comfy: Community and feature similarity guided rewiring. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b2c892312af07f8a77afbeed188391f4-Abstract-Conference.html.

Jake Topping, Francesco Di Giovanni, Benjamin P. Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Floriano Tori, Vincent Holst, and Vincent Ginis. The effectiveness of curvature-based rewiring and the role of hyperparameters in GNNs revisited. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ea7c7e886d48dd3a3a799819c8095012-Abstract-Conference.html.

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

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/58ae23d878a47004366189884c2f8440-Abstract.html.

A Exact Balanced Forman Curvature

We use the exact unweighted Balanced Forman curvature (BFC) of Topping et al. (2022) on the simple undirected graph underlying the message-passing graph. This is deliberate: BFC is defined for the undirected geometry, while the computational graph may store both directions of an edge.

For an undirected edge $\{u, v\}$ with degrees d_u, d_v, let $\Delta_{uv} = N(u) \cap N(v)$ be the common neighbors (triangles), $T_{uv} = |\Delta_{uv}|$, $d_{\max} = \max(d_u, d_v)$ and $d_{\min} = \min(d_u, d_v)$. Let Q^u_{uv} and Q^v_{uv} count the endpoint-side incidences of non-triangulated four-cycles through $\{u, v\}$, and let γ_{uv} be the maximum degeneracy, i.e. the maximum number of opposite nodes paired with a single endpoint-side neighbor among those four-cycles. Our implementation evaluates

$$\kappa^{\mathrm{BFC}}_{uv} = \frac{2}{d_u} + \frac{2}{d_v} - 2 + \frac{2T_{uv}}{d_{\max}} + \frac{T_{uv}}{d_{\min}} + \mathbb{1}[\gamma_{uv} > 0] \frac{Q^u_{uv} + Q^v_{uv}}{d_{\max}\gamma_{uv}}, \tag{13}$$

with the paper's special case $\kappa^{\mathrm{BFC}}_{uv} = 0$ when $\min(d_u, d_v) = 1$. The implementation has canonical self-tests: $C_3 = 1.5$, $C_4 = 1$, $C_5 = 0$, and $K_4 = 4/3$.

For a directed/bidirected message edge (u, v), we map the BFC of the underlying undirected edge $\{u, v\}$ back to the directed edge. This keeps the geometry faithful to the curvature definition without forcing a particular storage convention for message passing.

B Candidate Generation and Complexity

We first build the set of directed two-hop non-edges. For each pair (u, v), the middle-node set is

$$\mathcal{P}_{uv} = \{m : (u, m) \in E, (m, v) \in E\}.$$

The code computes this set through sparse adjacency multiplication and removes self-loops and existing one-hop edges. The topology score in Eq. (5) is then computed from exact BFC bottleneck severities on the two original path edges. We sort candidates within each source node and retain at most k.

The message-passing graph after candidate construction has at most $|E| + k|V|$ directed edges before optional symmetrization, rather than all possible two-hop pairs. This bound controls the cost paid by every later GATv2 layer. Exact BFC preprocessing is more expensive than a degree-only proxy because it counts local triangles and non-triangulated four-cycles. We compute it once per input graph/candidate state rather than inside every optimization step.

The budget-matched Uniform-2Hop control samples at most k two-hop non-edges per source from the same eligibility rule but does not use the topology ranking. Thus its edge budget is directly comparable to CARAT.

C Hard Consolidation Variant

The canonical method in the paper is CARAT, which keeps the candidate set soft throughout training. CARAT-H adds one irreversible transition after warm-up.

PyG GAT attention is normalized over incoming edges of each target. For attention α_{ij} and target in-degree d_j, we define

$$r_{ij} = d_j \alpha_{ij}. \tag{14}$$

Under uniform incoming attention, $r_{ij} \approx 1$. We therefore use

$$U_{ij} = g_{ij} r_{ij} \tag{15}$$

as a relative utilization score and track its exponential moving average

$$\bar{U}^{(t)}_{ij} = \mu \bar{U}^{(t-1)}_{ij} + (1 - \mu) U^{(t)}_{ij}. \tag{16}$$

The pruning rule operates only on shortcut edges. Since utilization is positive and skewed, the implementation works in log space. For shortcut scores $s = \log(\bar{U} + 10^{-8})$, let $m = \mathrm{median}(s)$, $\mathrm{MAD} = \mathrm{median}(|s - m|)$ and $\hat{\sigma} = 1.4826\,\mathrm{MAD}$. A shortcut is kept when

$$s \geq m - z\hat{\sigma}, \tag{17}$$

with $z = 1$ in all final runs. A safety rule ensures that no more than 50% of shortcut edges are deleted in one transition. Original edges are always retained. Node tasks use a 100-epoch warm-up; LRGB uses 20 epochs. After consolidation the best-checkpoint state is reset, so a reported CARAT-H model cannot silently restore a pre-pruning checkpoint.

D Datasets and Processing

Table 4: Node-classification datasets. Edge counts are message-passing edges after the final preprocessing convention. Chameleon-F and Squirrel-F are the filtered versions of Platonov et al. (2023); their undirected input edges are stored bidirectionally for PyG message passing.

Dataset	Nodes	Message edges	Features	Classes
Cora	2,708	10,556	1,433	7
Citeseer	3,327	9,104	3,703	6
Chameleon-F	890	17,708	2,325	5
Squirrel-F	2,223	93,996	2,089	5

Cora and Citeseer use standard Planetoid masks. For Chameleon-F and Squirrel-F we use the filtered data and provided split masks. The final results use 10 runs; stochastic initialization/dropout varies for Cora/Citeseer, while the filtered datasets also cycle through their provided splits in the experiment runner.

Peptides-func and Peptides-struct come from LRGB (Dwivedi et al., 2022). The benchmark contains 15,535 peptide graphs. Peptides-func is a 10-label graph-classification task evaluated by Average Precision; Peptides-struct is an 11-target graph-regression task evaluated by MAE. The model retains LRGB atom/bond encoders. Added shortcuts receive a learned shortcut-edge token, so they are not assigned a fabricated chemical bond type.

E Hyperparameters and Training Protocol

Table 5: Final hyperparameters. All variants within a task family share the same backbone and optimizer settings.

Setting	Node classification	LRGB Peptides
GATv2 layers	2	4
Hidden dimension	64	128
Attention heads	4	4
Compatibility dimension	32	32
Dropout	0.6	0.2
Optimizer	Adam	Adam
Learning rate	5×10^{-3}	10^{-3}
Weight decay	5×10^{-4}	10^{-5}
γ (bridge)	0.01	0.01
δ (sparsity)	0.001	0.001
Warm-up (hard variant)	100	20
Maximum epochs	600	100
Early-stop patience	50	20
Candidate top-k	8 (validation selected)	3 (fixed)
Train / eval batch	full graph	32 / 64 graphs
Runs	10	5

For AFP-enabled variants, early stopping and checkpoint selection are gated by the structural transition. At the transition, the best validation value, checkpoint, epoch, and patience counter are reset. This prevents a final "hard" model from being evaluated with a pre-pruning state. For non-AFP node models, a common minimum horizon is applied so that the comparison does not reward a baseline simply for terminating before the structural warm-up used by another model.

Model-name correspondence. The paper renames the implementation for clarity:

Paper name	Result/code key
GATv2	`gatv2_plain`
Uniform-2Hop	`gatv2_uniform_2hop`
Topology-only	`fecasa_v3_topology_only`
CARAT	`fecasa_v3_without_afp`
CARAT-H	`fecasa_v3_full`

The implementation key `without_afp` is the canonical soft method in this paper because the final validation/results show that hard consolidation is not consistently beneficial.

F Validation-Only Candidate-Budget Selection

The node candidate budget was selected from $k \in \{3, 5, 8\}$ before the final test table. The selection score was the mean validation accuracy of the hard model on Chameleon-F and Squirrel-F. Test metrics were not used. The stored selection record is:

k	Chameleon-F val.	Squirrel-F val.	Mean
3	44.27	37.36	40.82
5	45.04	37.98	41.51
8	**45.56**	**38.35**	**41.96**

Thus $k = 8$ was frozen for the final node table. Figure 4 visualizes the same validation-only sweep.

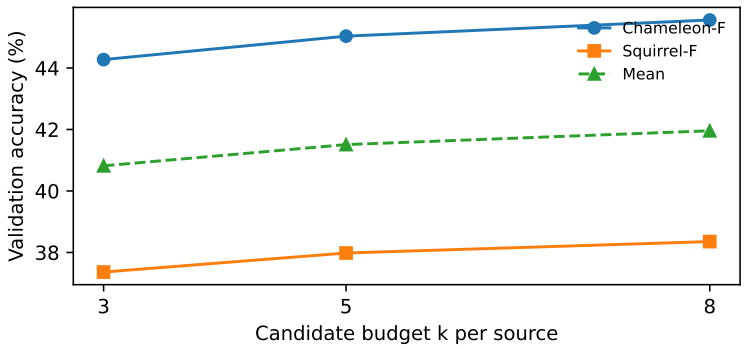


Figure 4: Validation-only candidate-budget sweep used before final node test evaluation.

G Shortcut Statistics

Table 6 reports the mean number of shortcut edges in the final node runs. Uniform-2Hop keeps its complete budget. CARAT is soft and therefore retains all candidates in the computational graph but learns continuous gates. Topology-only and CARAT-H perform hard consolidation after warm-up.

Table 6: Mean shortcut count after the model's final structural state.

Method	Cora	Citeseer	Chameleon-F	Squirrel-F
Uniform-2Hop	28,356	19,217	12,772	34,326
Topology-only	24,340	16,421	10,110	26,222
CARAT	29,018	19,658	12,190	31,262
CARAT-H	24,203	16,258	10,033	26,250

For CARAT, the mean soft gates are 0.780 (Cora), 0.770 (Citeseer), 0.649 (Chameleon-F), and 0.730 (Squirrel-F). The relative utilization score $U = g\, d_j \alpha$ is not a probability and may exceed one; a

value of one corresponds roughly to a unit gate with uniform incoming attention. Figure 5 shows the dataset-level means.

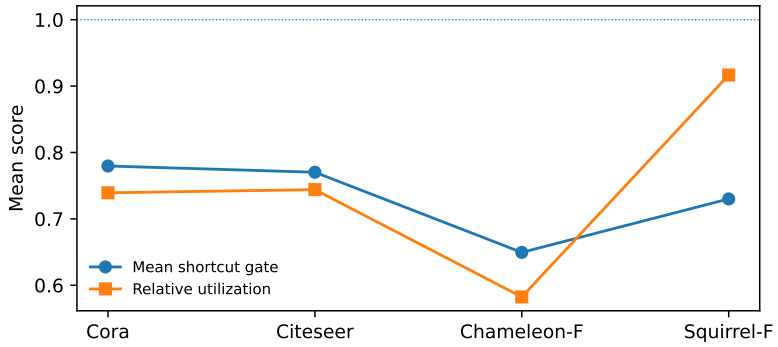


Figure 5: CARAT shortcut gate and relative utilization means. The horizontal reference at one corresponds to unit gate and uniform incoming attention.

H Paired Statistical Checks

Because each model family was run with matched seeds, we additionally compute two-sided paired t-tests on seed-level test metrics. These tests are descriptive checks rather than a substitute for effect size, especially on LRGB where $n = 5$.

For node classification, CARAT vs. GATv2 gives mean paired changes of -4.05 points on Cora ($p \approx 1.1 \times 10^{-5}$), $+1.37$ on Citeseer ($p \approx 6.0 \times 10^{-5}$), -1.09 on Chameleon-F ($p \approx 0.34$), and $+0.53$ on Squirrel-F ($p \approx 0.42$). Uniform-2Hop is significantly below GATv2 on Squirrel-F (mean -1.97 points, $p \approx 0.0059$), which is the main evidence that uncontrolled shortcut choice can be harmful even under the same per-source budget.

On Peptides-func, CARAT improves over GATv2 by $+0.0665$ AP ($p \approx 1.4 \times 10^{-7}$) and Uniform-2Hop by $+0.0633$ ($p \approx 7.2 \times 10^{-7}$). CARAT's additional mean gain over Uniform-2Hop is $+0.0032$ AP ($p \approx 0.048$); with only five seeds we treat this margin cautiously. On Peptides-struct, CARAT improves over GATv2 by about 0.0545 lower MAE ($p \approx 1.4 \times 10^{-4}$), whereas the difference between CARAT and Uniform-2Hop is not statistically clear ($p \approx 0.58$).

These checks reinforce the main text's interpretation: the large, reliable effect on Peptides is shortcut access itself; the extra quality-control effect is smaller.

I Implementation Pseudocode

1. Compute exact BFC on the underlying simple undirected geometry and map it to message edges.
2. Convert negative BFC to bottleneck severity $b \in [0, 1]$.
3. Enumerate eligible two-hop non-edges and compute t_{uv} by Eq. (5).
4. Keep at most top-k candidates per source; preserve all original edges.
5. At every forward pass, compute directional compatibility c_{uv}, structural score s_{uv}, and gate g_{uv} for shortcut edges. Set $g = 1$ on original edges.
6. Supply $[g, \mathbb{1}_{\text{original}}]$ as structural edge attributes to GATv2 and optimize task loss plus shortcut-only bridge/sparsity regularization.
7. For CARAT-H only, track relative utilization during warm-up, apply one robust shortcut-only consolidation transition, reset best-checkpoint state, and continue task training on the consolidated graph.