Calibrated Bounded Message Passing for Robust Graph Learning

Anonymous authors
Paper under double-blind review

Abstract

Message-passing graph neural networks learn node representations by aggregating information from their neighbours. However, widely used layers construct messages that depend linearly on neighbouring states, allowing a single anomalous or injected node to arbitrarily shift the aggregate. We introduce Calibrated Bounded Message Passing (CBMP), which normalises node inputs on both the message and residual streams and applies a calibrated saturating map before neighbourhood aggregation. Per-channel scales are estimated on the clean training graph and frozen, anchoring saturation thresholds to typical clean messages. We prove that changes in a node's aggregate scale with the total aggregation weight assigned to corrupted neighbours, and derive injection bounds under explicit degree conditions. Across five node-classification graphs with ten paired seeds (five on ogbn-arxiv), CBMP matches or exceeds GCN's clean performance and consistently improves robustness over GCN to feature outliers and node injection. Under degree-targeted, feature-adaptive injection, it gains 4–66 percentage points over GCN and outperforms the GRB LayerNorm defence on both heterophilous graphs and ogbn-arxiv while matching it on the other two graphs. Ablations isolate the benefit of bounding each incoming message and normalising both streams. These results establish pre-aggregation message control as an effective design principle for robust graph learning.

1 Introduction

In the message-passing layers that dominate practice the message is linear in the neighbour's state, and the nonlinearity comes only after the neighbourhood sum (Kipf & Welling, 2017; Hamilton et al., 2017; Brody et al., 2022). A single neighbour whose features lie far from the training distribution can therefore move the aggregate by an arbitrary amount, which is what node-injection attacks exploit (Zou et al., 2021; Chen et al., 2022; Zheng et al., 2021). Robust aggregators replace the mean by a location estimate with a breakdown point (Geisler et al., 2020; 2021; Hou et al., 2024). They act after the messages are formed, cost several forward passes, and still admit messages of unbounded size. Normalisation defences (Zheng et al., 2021) instead place a LayerNorm in front of every layer, which is cheap and effective against injection. The message nevertheless remains linear in the normalised input, so a neighbour's influence is limited only through the learned weights, and which component of such a scheme carries the gain has not been isolated.

We introduce Calibrated Bounded Message Passing (CBMP), a message-passing layer that controls each neighbour's contribution before aggregation. CBMP couples two complementary mechanisms: per-node normalisation on both the message and residual streams controls the scale entering the layer, while a calibrated saturating message map bounds each sender's contribution. The per-channel calibration statistic is estimated on the clean training graph and frozen, anchoring the saturation threshold to the clean operating regime. This design yields influence bounds governed by the aggregation weight assigned to corrupted neighbours and explicit injection bounds under degree-dependent conditions. Empirically, the bounded message separates CBMP from an equally normalised linear message under degree-targeted

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

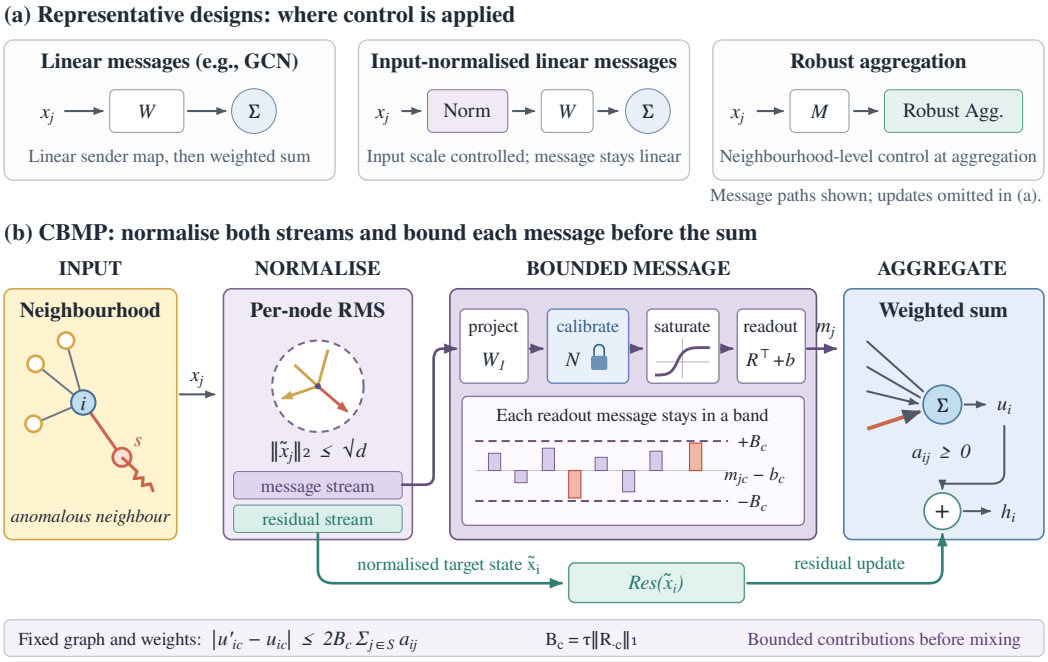

Figure 1: Where a defence applies its control. (a) Three representative designs, drawn as message paths with the update omitted: a linear sender map followed by the weighted sum, as in GCN and its relatives; the same map with the input scale normalised first, which leaves the message linear in the normalised input; and a robust aggregator, which acts on the neighbourhood once the messages have already been formed. (b) CBMP normalises the node state once and feeds that same normalised state to the message and the residual stream. The message is projected, rescaled by a per-channel statistic estimated on the clean training graph and then frozen (padlock), saturated and read out, so every message lands inside a fixed band $\pm B_c$ before it is summed. On a fixed graph with non-negative weights a corrupted set S can therefore shift a node's aggregate by at most $2B_c \sum_{j \in S} a_{ij}$ (Proposition 1). Curves and bars are schematic.

adaptive injection, while the two-stream normalisation preserves clean predictive performance and strengthens robustness across datasets.

This work makes three contributions. We introduce CBMP, a message-passing layer that combines two-stream input normalisation with a calibrated bounded message function, constraining individual neighbour contributions before aggregation with modest computational overhead. We establish per-node influence bounds for feature replacement and node injection, explicitly connecting robustness to corrupted aggregation-weight mass and graph degree. And through a controlled evaluation on five node-classification graphs, we show that CBMP preserves clean accuracy relative to GCN, improves robustness to anomalous features and adaptive injection, and isolates message boundedness and two-stream normalisation as the key mechanisms through targeted ablations.

2 Related work

A message-passing layer computes $u_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} a_{ij} M(h_j)$ followed by an update (Gilmer et al., 2017); GCN, GraphSAGE and attention layers use a linear M, so one neighbour's influence on u_i is unbounded. A message MLP with a bounded hidden activation, $M(h_j) = R^\top \phi(W_1 h_j)$, bounds each message, but nothing in standard training pushes the pre-activations into the saturating region: under the usual scaffold a tanh message MLP keeps its first-layer pre-activations well inside the linear range on four of five graphs (mean

$|z|$ 0.004–0.21; Appendix N), because rescaling the pre-activation down and the readout up leaves the loss nearly unchanged. The calibration of Section 3 removes that degree of freedom; it follows the inference-time statistics of Batch Renormalization (Ioffe, 2017) but is set on the clean training graph only and never re-estimated.

Normalisation has already been used as a defence. The Graph Robustness Benchmark (Zheng et al., 2021) reports that an affine LayerNorm applied to the raw features and to every layer's input makes GCN, GraphSAGE and GAT markedly more robust to injection and ships it as an option; GraphSAGE L2-normalises every layer's output; MsgNorm (Li et al., 2023) rescales the aggregated message; LipschitzNorm (Dasoulas et al., 2021) normalises attention scores. We reproduce the GRB scheme in our scaffold as a baseline and isolate its factors (which stream, which layer, affine or not, raw features or operator input; Appendix O); the input normalisation of CBMP is the statistic-free RMSNorm variant of that ingredient, and we say so wherever it appears.

A second family replaces the aggregation itself. Soft Medoid and Soft Median (Geisler et al., 2020; 2021) replace the mean by a robust location estimate; RUNG (Hou et al., 2024) solves an M-estimation problem in the aggregation; AirGNN and ElasticGNN (Liu et al., 2021a;b) shrink the residual with ℓ_{21}/ℓ_1 smoothers; RobustGCN and GNNGuard (Zhu et al., 2019; Zhang & Zitnik, 2020) act on edge weights. All act on the aggregate or on the weights with statistics of the (possibly perturbed) neighbourhood; a bounded message acts one step earlier with frozen clean-graph statistics, but limits nothing about the fraction of corrupted neighbours. Injection attacks (Zou et al., 2021; Chen et al., 2022) and adaptive evaluation (Mujkanovic et al., 2022) shape our attack protocol; certified robustness (Zügner & Günnemann, 2019) is a different threat model. Soft Median and an adapted AirGNN are compared in the shared scaffold; RUNG's code base and the official TDGIA/GRB attacks were not run (Section 7).

3 Calibrated Bounded Message Passing

Every model in this paper uses the same architecture: an encoder $\mathrm{Linear}(d_{\mathrm{in}} \to H)$ with dropout, $L = 2$ layers of the form $h_i \leftarrow u_i + \mathrm{Res}(x_i)$ with a linear residual map, LayerNorm, ELU and dropout between layers, and a last layer that maps to the classes; only the operator that produces u_i differs. The scaffold decides much of the accuracy of classic GNNs (Luo et al., 2024), so it is held fixed.

Let $x_j \in \mathbb{R}^d$ be the input of the layer at node j and $a_{ij} \geq 0$ the symmetric GCN weights with self-loops. CBMP computes

$$\begin{aligned} \tilde{x}_j &= \frac{x_j}{\sqrt{\mathrm{mean}_c x_{jc}^2 + \epsilon}}, & m_j &= R^\top \phi_\tau\big(N(W_1 \tilde{x}_j)\big) + b, \\ u_i &= \sum\nolimits_{j \in \mathcal{N}(i) \cup \{i\}} a_{ij}\, m_j, & h_i &= u_i + \mathrm{Res}(\tilde{x}_i), \end{aligned} \tag{1}$$

with $W_1 \in \mathbb{R}^{H_m \times d}$, $R \in \mathbb{R}^{H_m \times C}$ ($R_{\cdot c}$ is column c), $\phi_\tau(z) = \tau \tanh(z/\tau)$ applied coordinate-wise, and $N(z)_c = z_c/\sqrt{r_c + \epsilon}$ the calibrated per-channel scale defined next. Three choices define the layer. (i) Input normalisation on both streams. $\tilde{x}$ has no affine parameter, so $\|\tilde{x}_j\|_2 \leq \sqrt{d}$ for every node; the same $\tilde{x}$ enters the message function and the residual map. This is the GRB / RMSNorm ingredient; normalising the message stream alone is not enough (Section 6.2). (ii) Bounded message. Each hidden coordinate of ϕ_τ lies in $[-\tau, \tau]$, so $|m_{jc} - b_c| \leq B_c := \tau \|R_{\cdot c}\|_1$ whatever the input; $\tau = 1$ throughout. (iii) Calibrated operating point. r_c is the mean square of $z_c = (W_1 \tilde{x})_c$ over the nodes of the clean training graph, set during a calibration pass (evaluation mode, dropout off, no labels) immediately before each epoch's validation pass and frozen otherwise; it is stored with the checkpoint, and at test time, under every corruption and attack, it is fixed, so a corrupted graph cannot move it. At inference N folds into W_1, so CBMP is in the same function class as the uncalibrated layer; what calibration changes is the operating point that training reaches, by removing the scale degree of freedom through which gradient descent otherwise drives $|z| \ll \tau$ and makes the bound vacuous. Scale parameters (LayerNorm affine) are excluded from weight decay. Pseudo-code, the calibration order and the checkpoint contents are in Appendix H.

The layer costs one per-node normalisation per layer, one calibration pass per epoch (at most one forward pass), no parameter beyond the message MLP, and a single sender-side evaluation of the message per node (the message depends on the sender only, so it is never materialised per edge); measured in Section 6.4.

The three choices address different routes by which one neighbour can dominate a node's aggregate. Per-node normalisation removes the scale of the incoming state, so no neighbour enters the layer with an arbitrarily large magnitude; because the message and the residual map read the same normalised state, the residual path cannot reintroduce the scale that the message path has just removed. The saturating map then caps each sender's contribution before summation, which is what the linear messages of GCN and of LayerNorm defences leave unconstrained: their influence is limited only through the learned weights. Calibration makes that cap meaningful. Training is free to shrink the pre-activation and enlarge the readout without changing the loss, which drives the operator into the linear region of ϕ_τ and makes the bound vacuous; pinning the pre-activation scale to a clean statistic removes that degree of freedom. The statistic must come from clean data and stay fixed, since a threshold re-estimated on a perturbed graph would move with the attack. Concretely, each epoch runs a calibration pass over the clean training graph in evaluation mode, with dropout off and no labels, which writes r; the subsequent gradient step and the validation pass both use that frozen r; the value is saved in the checkpoint; and at test time, under every corruption and attack, r is never recomputed. Aggregation itself is left untouched: the weights stay non-negative and the message is a function of the sender alone, so the layer composes with standard sparse message passing. Alternatives that constrain the final message instead of the pre-activation, and a gate on the aggregation weights, are evaluated in Appendices C and D.

4 What the constraints guarantee

Proposition 1 (Per-node influence bound). Fix the graph, the non-negative aggregation weights a_{ij}, the parameters and the calibrated statistic, and all inputs except those of a node set S; let u_i, u_i' be the outputs of equation 1 before and after an arbitrary change of the features of the nodes in S. Then for every channel c, $|u'_{ic} - u_{ic}| \le 2\tau \|R_{\cdot c}\|_1 \sum_{j\in S} a_{ij}$.

(Proof in Appendix B.) The constant is attained in the relaxed space of independently chosen bounded messages; for a fixed network it is an upper bound only, never violated on trained checkpoints (Appendix F). Its content is the coupling to the operating point: it is informative only when the clean messages are of the order of $\tau\|R_{\cdot c}\|_1$, and calibration pins the pre-activation scale, not $\|R\|$, so this is measured (Appendix N). No such bound exists for the linear message of GCN or of the GRB scheme, whose influence is bounded only through $\|W\|$: with $\|\tilde{x}\|_2 \le \sqrt{d}$ a linear message $W\tilde{x}_j$ satisfies $|u'_{ic} - u_{ic}| \le 2\sqrt{d}\,\|W_{c\cdot}\|_2 \sum_{j\in S} a_{ij}$ (corollary in Appendix B), a constant no normaliser trains away but which weight decay must control; this is also the bound that CBMP's residual stream obeys.

Proposition 2 (Injected nodes). Fix the parameters, the calibrated statistic and the features of the existing nodes, and add a set L of new nodes with arbitrary features and edges. Let a_{ij} and a'_{ij} be the aggregation weights of node i before and after the injection and $B_c = \tau\|R_{\cdot c}\|_1$. Then for every channel c

$$|u'_{ic} - u_{ic}| \;\le\; B_c\Big[\sum_{j\in\mathcal{N}(i)\cup\{i\}} |a'_{ij} - a_{ij}| + \sum_{\ell\in L\cap\mathcal{N}'(i)} a'_{i\ell}\Big]. \tag{2}$$

With mean normalisation ($a_{ij} = 1/(d_i+1)$), k injected nodes attached to i only and the old neighbours' degrees unchanged, the bracket is $2k/(D+k)$ with $D = d_i + 1$; with symmetric normalisation under the same conditions it is $\frac{k}{D(D+k)} + \big(\frac{1}{\sqrt{D}} - \frac{1}{\sqrt{D+k}}\big)\sum_{j\in\mathcal{N}(i)} \frac{1}{\sqrt{d_j+1}} + \sum_\ell \frac{1}{\sqrt{(D+k)(d'_\ell+1)}}$. For any other injection pattern the general bracket must be evaluated with the post-injection degrees of every node.

(Proof in Appendix B.) Under the stated conditions the influence of k injected nodes decays like $2k/(D+k)$ under mean normalisation and only like $kD^{-1/2}$ under the symmetric one; it

is largest when the target's neighbours have low degree, does not decay if the budget grows with D, and is not a function of k and D alone once injected nodes attach to the neighbours (three-node example in Appendix B).

Both statements characterise a single layer's aggregate under fixed non-negative weights and a fixed calibration statistic. The layers that follow have finite but activation-dependent Lipschitz constants, so the bounds do not by themselves determine logits; Section 6.4 evaluates how far this single-layer control carries to end-to-end predictions.

Bounding each message caps the dynamic range of the neighbourhood sum, so information carried by the absolute magnitude of a neighbourhood travels through the residual path rather than the message path. Section 7 reports where this matters.

5 Experimental protocol

Hypotheses, models, datasets, corruption grids, seeds, statistics and decision rules of the first batch were frozen before any test evaluation (a protocol hash unlocks test evaluation; tuning code cannot read a test mask); later batches (normalisation controls, attack audits) were specified with their dates and run at test stage. CBMP's final form was developed on the training and validation nodes of Computers and Roman-empire only (two recorded rounds, no test metric computed), frozen, and evaluated once; the selection rule and its one relaxation are in Appendix D.

We evaluate on Amazon-ratings, Roman-empire and Tolokers (Platonov et al., 2023) (raw features, official splits), ogbn-arxiv (Hu et al., 2020) (official split) and Amazon-Computers (row-normalised features, frozen random 60/20/20 splits). Transductive: every model sees all node features; only training labels are used.

All models share the scaffold. CBMP and the unbounded-message ablation are tuned with the same 12-trial grid (learning rate, weight decay, dropout; validation only) as every tuned family; other ablations inherit their family's values. Baselines: GCN, GraphSAGE, GATv2, MLP, the unbounded message MLP with the same parameters (with and without input normalisation), GCN with input RMSNorm on the message stream or on both streams, GCN with input LayerNorm, the GRB LayerNorm scheme reproduced in the scaffold (affine LayerNorm on the raw features and every layer's input, single stream), per-message RMSNorm, final-message clipping, Soft Median (Geisler et al., 2021) with and without input RMSNorm, and an adapted AirGNN (Liu et al., 2021a); adapted or reproduced baselines are labelled so in every table.

Test nodes are never corrupted. Feature scaling (×2–16) and additive noise (1–8× RMS) on 5% or 10% of the other nodes; injection of 1% or 5% nodes with ten random edges and extreme features inside (×1) or beyond (×4) the training range; five corruption seeds. Adaptive attacks on fixed checkpoints, white box, worst case over true-gradient, straight-through and transfer modes: feature PGD on 10% of the non-target nodes (ℓ_∞ radius ε×RMS, 50 steps, 3 restarts); injection of 5% nodes with optimised features under nested centred boxes $\mathcal{B}_k = [c - kr, c + kr]$, $k \in \{1, 4\}$, warm-started from the smaller budget, with random connections or connections to the lowest-degree targets under a strict cap of two injected neighbours per target (edge budget kept; surplus edges go to non-targets); a fixed-norm direction attack on 10% of the nodes. Every reported number is the metric of one feasible attack graph; feasibility is asserted per graph (Appendix I).

We use ten paired seeds (five on ogbn-arxiv); corruption results are averaged over corruption seeds and levels before seeds become the unit; paired-bootstrap 95% intervals of the mean difference, exact sign-flip p-values Holm-corrected within each pre-specified family (floor 1/16 with five seeds), and a pre-registered 0.5-point non-inferiority margin for clean accuracy (Appendix T).

Table 1: Clean performance and robustness under adaptive node injection (↑). Test metric in percent (ROC-AUC for Tolokers); means over 10 paired seeds, or 5 for ogbn-arxiv. Inj. uses random connections and Targ. connects to the lowest-degree targets with at most two injected neighbours per target. Full threat-model and optimisation details are given in Section 5; extended results are in Table 25. Best per column in bold, second underlined. Provenance: † reproduced in our shared scaffold; ‡ our implementation; § adapted to our scaffold.

model	Amazon-r.			Roman-e.			Tolokers			Computers			ogbn-arxiv		
	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑
CBMP	47.6	37.6	28.1	82.0	71.8	58.3	83.2	74.1	66.3	90.2	87.3	83.8	70.2	63.0	54.5
CBMP-Base (no input norm.)	48.6	32.5	20.5	81.2	65.9	42.6	82.6	72.7	64.3	90.8	81.6	66.7	70.3	57.7	40.0
linear message + norm.	46.3	34.6	24.4	80.9	69.3	52.6	82.7	73.1	65.4	90.7	87.2	82.6	69.9	62.6	54.3
GCN	47.1	15.4	13.3	80.1	41.0	4.0	82.1	70.6	62.3	90.3	62.1	17.6	69.4	43.5	17.9
GCN + RMSNorm (message)	47.3	33.7	21.4	79.6	67.1	47.9	82.5	73.6	66.0	91.1	85.3	76.3	69.8	62.3	54.0
GCN + RMSNorm (both)	46.6	34.8	24.8	80.4	69.4	53.5	82.9	74.6	68.0	90.3	87.7	84.5	69.7	62.6	54.7
GRB LayerNorm†	47.8	35.0	23.3	81.2	70.0	53.0	83.4	74.7	66.5	90.4	87.5	83.8	69.9	62.2	53.9
per-message RMSNorm	49.0	36.6	23.9	81.3	69.9	53.2	83.0	73.3	64.8	91.1	83.3	71.9	70.5	60.8	48.1
clip only + norm.	46.1	35.4	25.8	80.4	69.1	53.5	81.8	72.4	66.5	89.6	86.8	83.2	69.1	61.1	50.3
Soft Median‡	46.2	30.3	18.6	79.0	65.2	39.4	82.0	73.6	65.7	89.9	75.3	47.2	68.9	58.6	44.9
Soft Median + RMSNorm‡	45.3	37.7	30.5	78.3	68.6	52.8	82.0	74.7	64.9	90.6	88.0	83.0	68.1	62.5	55.3
GATv2	49.2	17.1	15.3	81.9	54.3	13.5	83.9	63.8	55.1	91.1	51.3	7.3	70.7	14.8	10.0
GraphSAGE	48.5	14.1	13.3	79.9	39.6	4.6	82.3	73.4	67.3	90.6	67.4	26.8	70.0	50.5	30.1
AirGNN§	42.5	23.6	12.7	78.8	41.1	6.1	80.1	72.8	65.4	87.2	82.5	70.0	70.3	40.0	22.2
MLP (no graph)	41.9	41.9	41.9	65.5	65.5	65.5	73.1	73.1	73.1	84.0	84.0	84.0	53.7	53.7	53.7

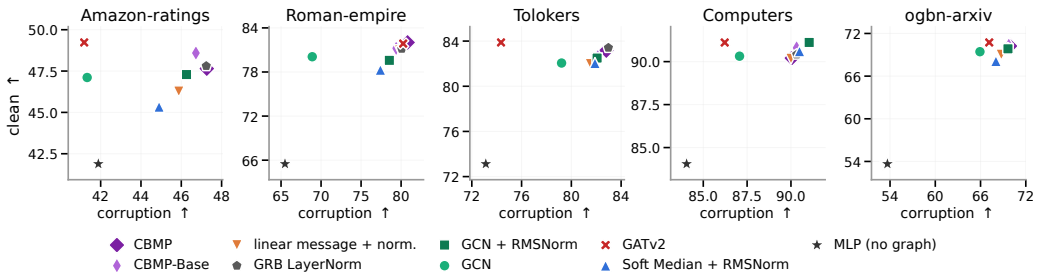

Figure 2: Clean test metric against mean corruption AUC (additive noise and feature scaling on 10% of nodes, injection of 5%; equal weights). Both axes are higher-is-better; the star is the no-graph MLP. Source data and plotting scripts are included in the supplementary material.

6 Results

All numbers are test-set values obtained once, at the best-validation checkpoint, under the frozen protocol: ten paired seeds per dataset, five on ogbn-arxiv. We report means with paired-bootstrap 95% intervals and Holm-corrected sign-flip tests; the full statistics, and per-seed values, are in Appendix T and the supplementary records. Datasets are always listed in the order Amazon-ratings, Roman-empire, Tolokers, Computers, ogbn-arxiv.

6.1 Clean accuracy and robustness to anomalous and injected neighbours

CBMP preserves clean accuracy and improves robustness on every graph. Clean, it is non-inferior to GCN on all five graphs (+0.5, +1.9, +1.1, −0.1, +0.8; every interval clears the pre-registered −0.5 margin and four exclude zero) and non-inferior to the GRB LayerNorm defence on all five (−0.2, +0.9, −0.3, −0.2, +0.3). GATv2 remains the most accurate model on four graphs, by 0.5–1.6 points. Averaged over additive noise, feature scaling and random in-range injection, CBMP keeps 47.3, 80.8, 82.8, 90.0, 69.9% against 41.3, 68.9, 79.2, 87.1, 65.9 for GCN, a gain of +2.9 to +11.9 points with every interval excluding zero (Figure 2, Table 25). Under random corruptions the two-stream normalisation does most of this work, and every model that has it sits close to the same ceiling.

The gap widens under adaptive injection, and most under degree-targeted attacks. With optimised injected features and random connections CBMP keeps 37.6, 71.8, 74.1, 87.3, 63.0%, against 15.4, 41.0, 70.6, 62.1, 43.5 for GCN; it exceeds the GRB defence on Amazon-

Table 2: Ablation (↑). Test metric in percent; clean, random-connection injection and degree-targeted injection, as in Table 1. Rows above the rule keep CBMP's two-stream input normalisation. All rows except CBMP and the tuned linear message use the message-MLP family's hyper-parameters. Every variant is defined in the text; the full set is in Table 27. Best per column in bold, second underlined.

model	Amazon-r.			Roman-e.			Tolokers			Computers			ogbn-arxiv		
	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑	clean ↑	inj. ↑	targ. ↑
CBMP	47.6	37.6	28.1	82.0	71.8	58.3	83.2	74.1	66.3	90.2	87.3	83.8	70.2	63.0	54.5
− calibration	47.1	37.2	27.7	83.0	72.4	59.3	82.5	73.2	66.6	89.1	86.8	83.7	69.1	61.1	49.8
+ budget	47.8	37.8	27.9	82.0	71.7	58.3	82.5	73.4	67.3	89.5	87.0	83.9	69.0	61.1	49.9
− calibration + budget	47.1	37.3	27.9	83.0	72.7	59.8	82.6	73.5	67.1	89.2	86.9	84.0	69.1	61.1	49.8
linear message (tuned)	46.3	34.6	24.4	80.9	69.3	52.6	82.7	73.1	65.4	90.7	87.2	82.6	69.9	62.6	54.3
linear + clip	46.1	35.4	25.8	80.4	69.1	53.5	81.8	72.4	66.5	89.6	86.8	83.2	69.1	61.1	50.3
message stream only	48.6	36.6	24.8	80.9	69.7	53.3	82.7	73.1	64.7	91.1	82.5	69.4	70.3	61.1	49.3
budget after the sum	47.9	37.8	28.0	82.0	71.8	58.0	82.6	73.6	67.4	89.5	86.9	83.6	69.0	61.0	49.6
no input norm (CBMP-Base)	48.6	32.5	20.5	81.2	65.9	42.6	82.6	72.7	64.3	90.8	81.6	66.7	70.3	57.7	40.0
no input norm + budget	48.7	35.4	22.4	81.1	66.1	43.1	82.7	72.8	64.6	90.9	83.0	71.3	70.4	59.9	44.8
no input norm, linear + clip	45.0	30.1	20.1	79.5	63.2	37.0	81.4	71.9	65.1	89.9	83.4	74.5	68.8	57.9	42.7
GRB LayerNorm†	47.8	35.0	23.3	81.2	70.0	53.0	83.4	74.7	66.5	90.4	87.5	83.8	69.9	62.2	53.9

ratings, Roman-empire and ogbn-arxiv (+2.6 [2.2, 3.0], +1.9 [1.6, 2.2], +0.8 [0.4, 1.2]) and matches it within 0.5 points elsewhere. When the attacker instead attaches two optimised neighbours to the lowest-degree targets, CBMP keeps 28.1, 58.3, 66.3, 83.8, 54.5%: above the GRB defence by +4.8 [3.6, 6.0] on Amazon-ratings, +5.3 [4.9, 5.7] on Roman-empire and +0.6 [0.1, 1.2] on ogbn-arxiv, and statistically tied on Tolokers (−0.1 [−0.9, 0.6]) and Computers (0.0 [−0.3, 0.4]); it is 7.6–17.2 points above the same operator without the input normalisation on every graph.

A fixed-norm directional attack separates the models in the same direction. Holding the norm of every attacked node exactly fixed and optimising only its direction on 10% of the nodes costs GCN 17–39 points and GATv2 19–66, against 4–18 for CBMP, which keeps 29.9, 66.5, 74.2, 86.2, 59.9%. That is +0.2 to +2.4 above an equally normalised linear message and −0.6 to +2.8 against the GRB defence.

6.2 Ablation: which constraint does what

We compare CBMP with six variants that each remove or move one constraint. − calibration removes the frozen per-channel statistic while retaining the input normalisation and the bounded activation. + budget adds an ℓ_2 projection $\Pi_\rho(m_j) = m_j / \max(1, \|m_j\|_2/\rho)$ to the final message, with ρ a frozen clean-graph quantile (Appendix C). Linear message replaces the calibrated saturating map with a parameter-matched linear message under the same two-stream input normalisation, and receives its own 12-trial tuning. Linear + clip adds the final-message projection to that linear variant. Message stream only normalises the input of the message function but leaves the residual stream unnormalised. Budget after the sum moves the projection from each message to the aggregate. Three mechanisms emerge.

Message boundedness, not the normalisation alone, produces the gain. Against the tuned linear message under the same two-stream normalisation, CBMP is ahead under degree-targeted injection by +3.7 [3.0, 4.6], +5.7 [5.1, 6.3], +1.0 [0.3, 1.6] and +1.3 [0.9, 1.6] on four graphs, and level on ogbn-arxiv (+0.2 [−0.2, 0.6]), with clean accuracy moving in the same direction (+1.4, +1.1, +0.5, −0.5, +0.3). Clipping the linear message recovers part of the difference but remains 2.4–4.8 points behind on Amazon-ratings, Roman-empire and ogbn-arxiv, so a saturating map at a calibrated operating point is not interchangeable with a norm cap.

Both streams have to be normalised. Normalising only the message stream leaves the residual path free to reintroduce the scale, and costs 3.3, 5.0, 1.6, 14.4 and 5.2 points under degree-targeted injection. This is also the factor that separates the GRB defence from a per-layer input normalisation alone (Appendix O).

Calibration and the explicit budget act as diagnostics rather than as sources of the gain. With normalised inputs, performance is insensitive to removing the calibration statistic (at most 0.4 points under degree-targeted injection, and −0.3 to +0.1 clean), indicating that the full model's empirical gains are driven primarily by message boundedness and two-stream

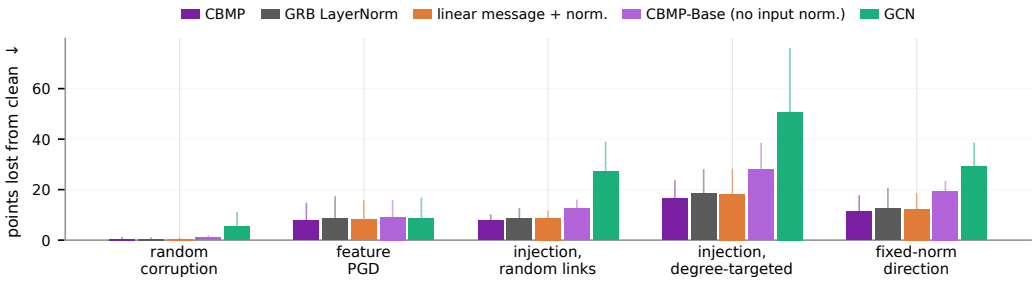

Figure 3: Robustness profile across threat models: points lost from the clean metric, averaged over the five graphs, with the per-graph range as a vertical line (lower is better). CBMP loses least under every threat model, and the margin over GCN widens as the attacker becomes more targeted. The underlying values are those of Table 1 and Table 25.

scale control. Calibration nevertheless fixes an interpretable clean-data operating point and remains important in the unnormalised diagnostic setting, where it is worth +7.4, −0.9, +0.7, +53.3 and +21.2 points. The explicit final-message budget provides no additional improvement in the full CBMP configuration (at most +0.3 points, and indistinguishable from the same projection applied after the sum), so we retain the simpler bounded-message design. Its gains in the unnormalised and linear diagnostics confirm that direct norm control can be useful when input scale is otherwise unconstrained: it is worth +1.9 to +4.8 points on the calibrated operator without input normalisation, and it lifts an unbounded linear message by +8.1 to +63.5 points. Depth, per-layer placement and the inter-layer normalisation are examined in Table 28; normalising the first layer's input alone is within 0.4 points of normalising every layer at two layers.

6.3 Robustness under adaptive attacks

Figure 3 summarises how far each model falls from its own clean accuracy under the five threat models, averaged over the five graphs. CBMP has the smallest mean drop in all five, and the ordering is stable: the three two-stream normalised models stay close under random corruption and feature PGD and separate under injection, where the unnormalised operators lose three to five times as much.

CBMP also leads under coordinated feature perturbations, by a smaller margin. Under ℓ_∞ PGD on the features of 10% of the non-target nodes ($\varepsilon = 0.2\times$RMS, worst case over true-gradient, straight-through and transfer modes) CBMP keeps 32.9, 70.4, 80.1, 86.9, 64.4%, above GCN by +2.5, +3.2, +0.8, +0.5 $[-0.1, 1.2]$, +1.0. The margins are smaller than under injection because a coordinated perturbation of many in-range neighbours stays inside the per-message bound: bounding a message limits what one neighbour can do, not what a tenth of the graph can do together.

Stronger attack budgets do not change the ordering. Raising the budget to 200 steps with 10 restarts, and switching between cross-entropy and margin objectives, moves CBMP's worst case by at most 1.1 points. Gradient-guided connection selection changes the degree-targeted numbers of CBMP, its uncalibrated variant, the linear message and the GRB defence by at most 0.9 points and reorders nothing; removing the per-target cap costs every model 2.9–3.6 points on Computers and nothing on Roman-empire. Every reported number corresponds to one attack graph that satisfies its budget, checked per graph (Appendix I), and the out-of-range feature budget is never weaker than the in-range one.

The graph structure itself remains a separate attack surface. Attaching two optimised neighbours to a low-degree target is strong enough that on Amazon-ratings, Roman-empire and Tolokers a graph-free MLP is the more robust predictor, although CBMP matches it on Computers (-0.2 $[-0.7, 0.2]$) and exceeds it on ogbn-arxiv ($+0.8$ $[0.6, 1.1]$), so the graph remains an asset on the two largest graphs (Figure 9, appendix). The directional attack likewise costs every graph model 15–19 points on the two heterophilous graphs, CBMP

included at 16–18. CBMP constrains what a neighbour may send, not which neighbours a node has.

6.4 Mechanism and cost

The single-layer bound tracks end-to-end behaviour. On trained CBMP checkpoints the layer-1 aggregate drift under $\times 16$ scaling of 10% of the nodes reaches 4–25% of the Proposition 1 bound and never exceeds it. The same perturbation flips 10–31% of GCN's and 4–36% of GATv2's test predictions, against 0.4–5.0% for CBMP. The bounded network does place more scale in the readout ($\|R\|_2$ of 3.4–6.5 against 1.1–3.4 for the linear message), which the bound charges for, yet its final messages remain smaller (99th-percentile norms 13–37 against 15–57); details in Table 26, appendix.

The overhead is the message MLP, not the constraints. On an idle GPU, CBMP's forward pass is 1.1–1.9× GCN's and 1.0–1.5× the GRB defence's; the once-per-epoch calibration pass costs one forward pass; peak training memory is 1.1–2.2× GCN's; and the parameter count is that of the message MLP (55–106k against 9–59k). For comparison, Soft Median costs 7–40× CBMP's forward pass, and attack evaluation scales with it. Full profiling is in Table 24, appendix.

7 Limitations

The bounds of Section 4 constrain a single layer's aggregate and have not been composed into a statement about logits; what we report end-to-end are measurements. The evaluation covers transductive node classification on five moderate-size graphs with a two-layer scaffold, so structure attacks, graph-level tasks and deep models remain to be tested. Comparisons against RUNG, AirGNN and the TDGIA and GRB attack suites use variants reproduced or adapted inside our shared scaffold rather than the authors' original pipelines, which bounds how finely those baselines can be compared. Exploratory revision batches preceded the frozen final specification; Appendix D records the test exposure, the validation-only selection of the final method, and the single documented relaxation of the selection rule.

Three regimes are harder for CBMP: perturbing many in-range neighbours at once, selecting neighbours by degree, and tasks whose labels depend on the absolute magnitude of a neighbourhood. The last is a direct consequence of the design. On synthetic graphs with magnitude- or count-dependent labels and no auxiliary feature, CBMP stays within -3.7 to $+4.4$ points of GCN and 7–17 points above a message-stream-only normalisation, because the residual path retains a route for node-specific magnitude, whereas the GRB defence discards it at the raw features and trails by 14–32 points unless a degree feature is supplied (88–92 with one). The cost of bounding is therefore graph- and label-dependent, and a model that must read absolute neighbourhood magnitude should keep an explicit degree or scale feature (Appendix P).

8 Conclusion

CBMP controls each neighbour's contribution before aggregation through a two-stream normalised architecture and a calibrated bounded message map. Across five node-classification graphs, it preserves clean performance relative to GCN while substantially improving robustness to anomalous features and adaptive node injection, with especially strong gains under degree-targeted attacks. The analysis connects robustness to corrupted aggregation-weight mass, and controlled ablations identify message boundedness and two-stream normalisation as the main empirical mechanisms. These results position pre-aggregation contribution control as a practical building block for robust graph learning, while motivating future work on compositional guarantees and adaptive structural defences.

Reproducibility statement

All code, the frozen protocol files with their hashes and amendments, the validation-only development records, tuning logs, raw JSONL records of every run (with data, split, config-

uration and code hashes), the run manifest, the attack feasibility ledger, summary scripts and figure source data are in the supplementary material. Datasets are public (PyTorch Geometric / OGB loaders); random splits are frozen and hashed. One command sequence rebuilds every number, table and figure from the raw records (REPRODUCE.md); a consistency script cross-checks every table, figure source and statistic before the PDF is built.

AI usage statement

The research question, the design of the method, the experimental design and every scientific decision in this paper are the authors' own. We used a generative AI assistant primarily for writing: drafting and revising the manuscript text. Under our specification and review, the same assistant also helped implement the operator, corruption and attack code and execute the training and evaluation scripts. No data or experimental result was generated by an AI tool: every number in the paper is produced by the released scripts from raw experiment records, and we verified the reported values against those records. The authors take full responsibility for the content.

References

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In International Conference on Learning Representations, 2022.

Spyridon Chavlis and Panayiota Poirazi. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. Nature Communications, 16:943, 2025. doi: 10.1038/s41467-025-56297-9.

Yongqiang Chen, Han Yang, Yonggang Zhang, Kaili Ma, Tongliang Liu, Bo Han, and James Cheng. Understanding and improving graph injection attack by promoting unnoticeability. In International Conference on Learning Representations, 2022.

George Dasoulas, Kevin Scaman, and Aladin Virmaux. Lipschitz normalization for self-attention layers with application to graph neural networks. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of PMLR, pp. 2456–2466, 2021.

Simon Geisler, Daniel Zügner, and Stephan Günnemann. Reliable graph neural networks via robust aggregation. In Advances in Neural Information Processing Systems, volume 33, 2020.

Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. In Advances in Neural Information Processing Systems, volume 34, 2021.

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 PMLR, pp. 1263–1272, 2017.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Advances in Neural Information Processing Systems, volume 30, pp. 1024–1034, 2017.

Zhichao Hou, Ruiqi Feng, Tyler Derr, and Xiaorui Liu. Robust graph neural networks via unbiased aggregation. In Advances in Neural Information Processing Systems, volume 37, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In Advances in Neural Information Processing Systems, volume 33, 2020.

Sergey Ioffe. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. In Advances in Neural Information Processing Systems, volume 30, pp. 1945–1953, 2017.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations, 2017.

Guohao Li, Chenxin Xiong, Guocheng Qian, Ali Thabet, and Bernard Ghanem. DeeperGCN: Training deeper GCNs with generalized aggregation functions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(11):13024–13034, 2023.

Xiaorui Liu, Jiayuan Ding, Wei Jin, Han Xu, Yao Ma, Zitao Liu, and Jiliang Tang. Graph neural networks with adaptive residual. In Advances in Neural Information Processing Systems, volume 34, pp. 9720–9733, 2021a.

Xiaorui Liu, Wei Jin, Yao Ma, Yaxin Li, Hua Liu, Yiqi Wang, Ming Yan, and Jiliang Tang. Elastic graph neural networks. In Proceedings of the 38th International Conference on Machine Learning, volume 139 of PMLR, pp. 6837–6849, 2021b.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic GNNs are strong baselines: Reassessing GNNs for node classification. In Advances in Neural Information Processing Systems: Datasets and Benchmarks Track, volume 37, 2024.

Felix Mujkanovic, Simon Geisler, Stephan Günnemann, and Aleksandar Bojchevski. Are defenses for graph neural networks robust? In Advances in Neural Information Processing Systems, volume 35, 2022.

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.

Xiang Zhang and Marinka Zitnik. GNNGuard: Defending graph neural networks against adversarial attacks. In Advances in Neural Information Processing Systems, volume 33, pp. 9263–9275, 2020.

Qinkai Zheng, Xu Zou, Yuxiao Dong, Yukuo Cen, Da Yin, Jiarong Xu, Yang Yang, and Jie Tang. Graph robustness benchmark: Benchmarking the adversarial robustness of graph machine learning. In Advances in Neural Information Processing Systems: Datasets and Benchmarks Track, 2021.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1399–1407, 2019. doi: 10.1145/3292500.3330851.

Xu Zou, Qinkai Zheng, Yuxiao Dong, Xinyu Guan, Evgeny Kharlamov, Jialiang Lu, and Jie Tang. TDGIA: Effective injection attacks on graph neural networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2461–2471, 2021. doi: 10.1145/3447548.3467314.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 246–256, 2019. doi: 10.1145/3292500.3330905.

A Exact reductions

Gated branches are a block-constrained message MLP. The gated M-branch operator computes $u_i = \sum_m \sigma(g_m) \odot \gamma_m \odot \sum_j a_{ij} \phi_\tau(N(W_m h_j)) + b$. Stacking $W_1 = [W_1 | \dots | W_M] \in \mathbb{R}^{d \times MC}$ and defining $R \in \mathbb{R}^{MC \times C}$ by $R_{(m,c),c} = \gamma_{mc}\sigma(g_{mc})$ and zero elsewhere gives equation 1 exactly; forward outputs, input gradients and parameter gradients (with $\partial/\partial g$ obtained from $\partial/\partial R$ by the chain rule through σ) agree to floating-point precision (unit test test_dgn_subset_of_mlp_message). The dense-readout message MLP therefore contains the branched operator; the extra freedom is the $MC(C-1)$ off-block entries of R.

Linear branches fold exactly. With $\phi = \mathrm{id}$ and static gates, $\sum_m \sigma(g_m) \odot \gamma_m \odot W_m^\top h = W_\star^\top h$ with $W_\star = \sum_m W_m \,\mathrm{diag}(\gamma_m \sigma(g_m))$: the branches add nothing to the function class (test_linear_fold_exact). Any effect of M in the linear regime is an optimisation effect of over-parameterisation, which is why we do not attribute robustness or accuracy to branching.

B Proofs, the symmetric-normalisation case and a counterexample

Both propositions fix the parameters, the calibrated statistic r and ϕ_τ; Proposition 1 also fixes the graph, Proposition 2 the features of the existing nodes (so that their layer-1 messages are unchanged).

Proof of Proposition 1. Each hidden coordinate of ϕ_τ lies in $[-\tau, \tau]$, so each message component $R_{\cdot c}^\top \phi_\tau(\cdot)$ lies in $\tau[-\|R_{\cdot c}\|_1, \|R_{\cdot c}\|_1]$; only messages from S change, and the triangle inequality gives the claim.

Proof of Proposition 2. Existing messages depend on the sender only and are unchanged, so $u_i' - u_i = \sum_j (a_{ij}' - a_{ij}) m_j + \sum_\ell a_{i\ell}' m_\ell$; bound each $|m_{\cdot c}|$ by B_c. Equality: $m_{jc} = B_c \,\mathrm{sign}(a_{ij}' - a_{ij})$ and $m_{\ell c} = +B_c$, which a fixed network need not realise.

Degree decay under symmetric normalisation. With $a_{ij} = ((d_i{+}1)(d_j{+}1))^{-1/2}$, $D = d_i{+}1$, k injected nodes attached to i only, and the old neighbours' degrees unchanged, the bracket of equation 2 is $\frac{k}{D(D+k)} + \big(\frac{1}{\sqrt{D}} - \frac{1}{\sqrt{D+k}}\big) \sum_{j \in \mathcal{N}(i)} \frac{1}{\sqrt{d_j+1}} + \sum_\ell \frac{1}{\sqrt{(D+k)(d_\ell'+1)}}$: the self-loop term (weight $1/D \to 1/(D+k)$, at most k/D^2), the old-neighbour term and the injected term (at most $k/\sqrt{2(D+k)}$ for leaves). For the old-neighbour term the mean value theorem gives $D^{-1/2} - (D+k)^{-1/2} \le \frac{k}{2} D^{-3/2}$ (the derivative of $t^{-1/2}$ is $-\frac{1}{2} t^{-3/2}$, largest in magnitude at $t = D$), and the sum has $d_i = D - 1 < D$ terms each at most $(\min_j d_j + 1)^{-1/2}$, so the term is at most $\frac{k}{2}\, D^{-1/2} (\min_j d_j + 1)^{-1/2}$. For fixed k all three terms therefore vanish at least like $k\, D^{-1/2}$, more slowly than the $2k/(D+k)$ of mean normalisation; the middle term is largest when the target's neighbours have low degree; if k grows with D nothing decays.

Two checks (tests/test_revision2.py). (1) Target 0 with one neighbour 1 (both degree one), messages $-B_c$; a leaf 2 attached to 0 with message $+B_c$. Post-injection weights $(1/3, 1/\sqrt{6}, 1/\sqrt{6})$ against $(1/2, 1/2)$: exact drift $2B_c/3$; general bracket $1/6 + 0.092 + 0.408 = 0.667$ (tight). Two simplified expressions used in earlier drafts, one rescaling the clean aggregate by $\sqrt{D/(D+k)}$ and one treating the self-loop like a neighbour, both give $0.592 B_c < 2B_c/3$ and are wrong. (2) The injected node attaches to neighbour 1 only: $k = 0$ for the target, yet its aggregate moves by $0.092 B_c$ because a_{01} changes from $1/2$ to $1/\sqrt{6}$; the decay statements therefore hold only under the stated degree conditions, and the general bracket must be used otherwise.

Corollary for the input-normalised linear message (proof). With $N(h) = h/\sqrt{\mathrm{mean}_c h_c^2 + \epsilon}$ over d coordinates, $\|N(h)\|_2 = \sqrt{d}\,\|h\|_2 / \sqrt{\|h\|_2^2 + d\epsilon} \le \sqrt{d}$. For a message $m_j = W N(h_j) + b$ the bias cancels in every difference, so for a fixed graph with non-negative weights and only the nodes in S changed, $\|u_i' - u_i\|_2 = \|\sum_{j \in S} a_{ij} W (N(h_j') - N(h_j))\|_2 \le \sum_{j \in S} a_{ij} \|W\|_2 \cdot 2\sqrt{d}$ and, per coordinate, $|u_{ic}' - u_{ic}| \le 2\sqrt{d}\,\|W_{c\cdot}\|_2 \sum_{j \in S} a_{ij}$. The constant is not trained down by the normaliser (there is no affine parameter) but it does depend on $\|W\|$, which weight decay controls; it is a one-layer statement on the operator input space (the encoder output in our scaffold), not a classification certificate.

Tightness caveat (Proposition 1). The constant $2\tau\|R_{\cdot c}\|_1$ is the supremum over independently chosen hidden coordinates in $[-\tau, \tau]$. A fixed network need not reach it: with $z_j = (h, h)$ and $R_{\cdot c} = (1, -1)$ the message component is identically zero while the bound is 4τ; with tanh or softsign the extreme points are approached only as $|z| \to \infty$. Numerical non-violation (Table 4) is evidence that the bound holds, not that it is tight.

Multi-layer. At the second layer the messages of every node adjacent to L change; equation 2 then applies only to the messages of unaffected nodes, and a stacked statement would need the first-layer drift of the affected neighbours (Proposition 1 with S = nodes adjacent to L) propagated through LayerNorm, ELU and the residual, whose Lipschitz constants depend on the clean activations. We report measured end-to-end drifts instead.

Three different statements. (i) Boundedness (Propositions 1, 2) holds for any bounded ϕ and any normalisation; it is a property of the architecture. (ii) Tightness: the constant $B_c = \tau\|R_{\cdot c}\|_1$ is not controlled by unit-RMS pre-activations; Table 4 shows the observed layer-1 drift is 10–20% of the bound for the message MLP and 55–96% for the single-unit operator. (iii) Prediction stability: neither bound concerns logits; the following LayerNorm rescales by the inverse standard deviation of its input, so a bounded layer drift does not imply a bounded logit drift. The degree-stratified results are empirical, not implied.

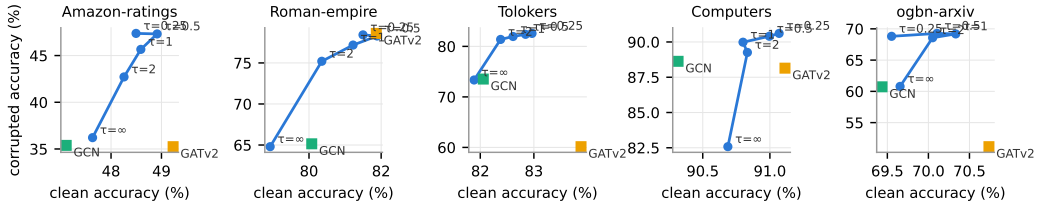

Figure 4: Clean vs. corrupted (additive 8× on 10% of nodes) test accuracy along the operating point τ of the bounded message MLP ($\tau = \infty$: linear message); GCN and GATv2 for reference.

C Alternative constraints and diagnostic ablations

Definition. $\Pi_\rho(v) = v/\max(1, \|v\|_2/\rho)$ applied to the final message $m_j = R^\top \phi_\tau(N(W_1\tilde{x}_j)) + b$ before the neighbour sum; ρ is the q-quantile ($q = 0.95$) of $\|m_j\|_2$ over the nodes of the clean training graph, set during the same calibration passes as the statistic r and frozen otherwise (stored in the checkpoint; a corrupted graph or injected nodes cannot move it). Messages inside the ball are unchanged; zero vectors stay zero; gradients pass through the scale factor (true-gradient attacks) or through the identity (straight-through mode).

Proposition 3 (Budgeted messages). With $\|m_j\|_2 \le \rho$ for every sender, a fixed graph with non-negative weights and only the nodes in S changed, $\|u'_i - u_i\|_2 \le 2\rho \sum_{j\in S} a_{ij}$; under the injection setting of Proposition 2, $\|u'_i - u_i\|_2 \le \rho\big[\sum_j |a'_{ij} - a_{ij}| + \sum_{\ell\in L\cap\mathcal{N}'(i)} a'_{i\ell}\big]$.

Proof. $u'_i - u_i = \sum_{j\in S} a_{ij}(m'_j - m_j)$ with $\|m'_j - m_j\|_2 \le 2\rho$; the injection case is the decomposition of the proof of Proposition 2 with $\|m\|_2 \le \rho$ in place of $|m_c| \le B_c$. Unlike $B_c = \tau\|R_{\cdot c}\|_1$, ρ does not depend on the learned R or W_1; it is a clean-data quantile. □

Role in the final design. On validation data (Computers, Roman-empire; three seeds; Appendix D) the budget improved the calibrated bounded message MLP without input normalisation on Computers (+5 to +9 points under targeted injection) but not on Roman-empire (−2 to +0.4), and on top of the input normalisation it added at most +0.6 points at any quantile; at test stage the same contrast is Table 2. A network can still move amplification into the readout and the residual path, which the budget does not touch; we measured $\|R\|_2$, the message norms and the budget trigger rate on the trained models (Table 21).

D Development record and selection rule

Development (validation only). Two recorded design rounds on the training and validation nodes of Computers and Roman-empire (three seeds, no test metric computed).

Round 1 compared the calibrated bounded message MLP with the final-message budget ($q \in \{0.9, 0.95, 0.99\}$), final-message clipping without calibration or bounded map, the budget without calibration, the budget after the sum, and the budget with input normalisation, against GCN, GCN + input RMSNorm, the GRB scheme, per-message RMSNorm and the MLP. Round 2 changed one thing, adding the per-node input RMSNorm on both streams, and varied the message function under it (calibrated tanh, uncalibrated tanh, linear, linear + clipping, with and without budget, message stream only). The validation numbers are in the supplementary material.

Selection rule and its relaxation. The rule written before the runs required (a) a clean validation loss of at most 0.5 points against the calibrated bounded message MLP on both graphs and (b) a targeted-injection gain larger than the seed standard deviation on both graphs. No candidate met both: the budget-only variants met (a) but not (b) on Roman-empire; every input-normalised variant met (b) by +16 to +19 points but lost 0.9–1.2 clean validation points on Computers (the GRB scheme loses 0.4 there). The clean criterion was relaxed to 1.0 point, on validation data and before any test run; this admitted the calibrated input-normalised variants and excluded the uncalibrated one (−1.2). Among the admitted variants the budget contrast was a tie at every q, so the budget was dropped and the minimal design frozen as CBMP.

Consensus gate on the aggregation weights. Before the validation-only development above, a candidate that re-weights each message by $g_{ij} = \exp(\kappa(\cos_{ij} - 1))$, with $\cos_{ij}$ the cosine between the message and the leave-one-out consensus of the neighbourhood and κ learned, was implemented on top of the per-message-normalised message MLP with input normalisation on both streams and run directly at test stage (ten seeds, five on ogbn-arxiv). Its test numbers were therefore seen before CBMP was frozen; they are reported in Table 3 for completeness and were not used to select CBMP. Under the v2 targeted attack the gate adds +1.8, −1.0, +0.7, +1.4, +2.2 points on Amazon-ratings, Roman-empire, Tolokers, Computers and ogbn-arxiv on top of the per-message-normalised, input-normalised message MLP (28.8, 58.6, 67.1, 83.7, 49.3 without it), and +3.6, −2.1, +1.6, +1.5, +2.3 on top of a linear message with the same normalisation, at a clean cost of 0.0–0.7 points; it is a data-dependent weight and therefore carries the drift term of Section 4. It was not carried forward because its evaluation did not follow the validation-first protocol and its gains are of the size of the seed spread on three graphs; it remains a candidate for future work.

Table 3: Consensus-gate candidate and its ablations (test %; v2 attack protocol; exploratory).

operator	Amazon-ratings						Roman-empire						Tolokers						Computers						ogbn-arxiv					
	clean	corr.	PGD	inj. ×1	targ. ×1	dir. att.	clean	corr.	PGD	inj. ×1	targ. ×1	dir. att.	clean	corr.	PGD	inj. ×1	targ. ×1	dir. att.	clean	corr.	PGD	inj. ×1	targ. ×1	dir. att.	clean	corr.	PGD	inj. ×1	targ. ×1	dir. att.
BMP (ours): (i)+(ii)+(iii)	48.3	47.8	34.3	40.2	30.6	32.1	82.3	80.8	70.3	71.8	57.6	66.5	82.3	82.2	79.7	74.8	67.8	73.6	90.3	89.4	87.4	87.9	85.1	87.2	70.2	68.9	63.5	62.5	51.5	59.7
BMP without gate: (i)+(ii)	48.4	47.9	32.8	38.6	28.8	29.9	82.5	81.0	70.5	71.9	58.6	66.7	82.6	82.4	80.2	73.6	67.1	73.4	90.3	89.4	86.7	87.2	83.7	86.1	70.4	69.2	63.3	61.0	49.3	57.4
(ii)+(iii), one stream	48.7	47.8	33.8	38.5	27.2	31.7	81.8	80.0	69.2	69.6	53.0	64.7	82.8	82.7	79.9	74.6	66.2	73.6	91.6	90.9	85.9	86.4	75.7	83.9	71.2	70.0	63.4	62.5	51.5	58.3
(ii) only: per-message-normalised MLP	49.1	48.3	32.6	36.6	23.9	29.8	81.8	80.1	69.3	69.9	53.2	64.8	82.9	82.6	80.1	73.3	64.8	74.6	91.5	90.8	83.1	83.3	70.8	78.4	71.2	70.2	63.1	60.8	48.1	55.3
BMP-linear (ours): (i)+(iii)	46.1	45.2	32.2	37.6	29.7	30.6	81.8	80.1	68.7	69.8	51.8	64.6	80.6	80.6	79.0	74.3	67.6	71.1	89.9	89.0	87.1	87.3	84.3	86.6	69.3	68.1	62.9	62.1	51.8	59.6
(i) only, linear message	46.5	45.5	30.4	35.6	26.1	28.1	81.8	80.0	68.6	69.7	53.9	64.3	81.3	81.0	79.2	71.9	66.0	70.6	89.9	89.1	86.5	86.6	82.8	85.5	69.4	68.2	62.5	60.5	49.5	57.4
GCN, GRB LayerNorm scheme	48.4	47.2	30.4	35.0	23.3	27.1	81.6	80.1	68.4	69.9	53.0	64.4	83.5	83.0	80.1	74.7	66.5	74.1	91.0	90.3	87.6	87.5	83.2	86.7	70.9	69.6	63.5	62.2	53.9	58.0
GCN + input RMSNorm (one stream)	47.8	46.3	31.0	33.7	21.4	27.3	80.2	78.5	66.1	67.1	47.9	62.0	82.6	82.1	79.4	73.6	66.0	73.5	91.7	91.0	85.1	85.3	75.5	82.7	70.7	69.6	63.5	62.3	54.0	59.0
Soft Median + input RMSNorm	46.0	44.9	31.3	37.7	30.6	30.3	79.1	77.4	65.8	68.6	52.8	62.4	82.1	81.9	78.5	74.7	64.9	74.5	91.1	90.5	87.9	88.0	82.4	87.8	–	68.0	62.5	62.5	55.3	60.2
GCN	47.1	41.3	30.3	15.4	13.3	15.1	80.1	68.9	67.3	41.0	4.0	57.8	82.1	79.2	79.3	70.6	62.3	64.9	90.3	87.1	86.5	62.1	17.3	54.6	69.4	65.9	63.4	43.5	17.9	30.8
GATv2	49.2	41.2	25.4	17.1	15.2	13.3	81.9	80.3	71.8	54.3	13.5	62.6	83.9	74.3	80.6	63.8	55.1	48.6	91.1	86.2	84.7	51.3	7.7	39.8	70.7	67.1	62.2	14.8	10.0	4.4
Soft Median	46.2	44.3	30.7	30.4	18.6	22.4	79.0	77.2	67.0	65.2	39.5	60.9	82.0	81.6	78.9	73.6	65.7	70.3	89.9	87.9	86.5	75.3	46.4	65.5	68.9	68.6	63.1	58.6	44.9	53.5
MLP (no graph)	41.9	41.9	41.9	41.9	41.9	41.9	65.5	65.5	65.5	65.5	65.5	65.5	73.1	73.1	73.1	73.1	73.1	73.1	84.0	84.0	84.0	84.0	84.0	84.0	53.7	53.7	53.7	53.7	53.7	53.7

Test-set exposure. The pre-registered batch evaluated the test set once per run after the hash unlock. The revision batches E5–E11 were specified with their dates but run at test stage; their numbers were seen by the authors before revision 4. CBMP (E12) is the first candidate developed on validation only.

E Degree dependence of the injection damage

Proposition 2 predicts, under mean normalisation, that the damage of injected nodes decays with the target's degree (and, under the symmetric normalisation used here, that it decays more slowly and depends on the neighbours' degrees). Figure 5 confirms it for every model, but from different levels: on Amazon-ratings, Computers and ogbn-arxiv the bounded model loses 2.8–2.9 points in the lowest-degree decile and about zero in the highest, whereas GCN and the linear message lose 7–22 points at low degree. Proposition 1 holds numerically

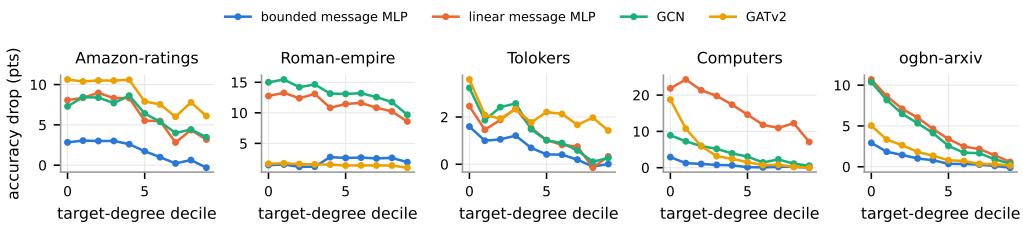

Figure 5: Accuracy drop of test nodes under injection of 5% in-range nodes, by decile of the target node's degree.

on the trained models: with 10% of the nodes scaled by 16, the largest observed layer-1 drift is 10–20% of the bound for the message MLP and 55–96% for the single-unit operator (Appendix F).

F Bound verification

Table 4 reports, for two trained seeds per dataset, the largest and mean ratio between the observed layer-1 drift $\|u_i' - u_i\|_\infty$ and the bound of Proposition 1 when 10% of the non-test nodes have their features scaled by 16. The ratio never exceeds one; it is far from one for the message MLP because $\|R_{\cdot c}\|_1$ over-counts the dense readout, and closer to one for the single-unit operator. On the synthetic graphs of Appendix P the largest ratio over all settings is 0.93. The degree-decile results of Figure 5 are the empirical counterpart of Proposition 2.

Table 4: Numerical check of Proposition 1 on trained models (scale ×16 on 10% of nodes).

dataset	model	max drift/bound	mean drift/bound	mean drift	mean bound
Amazon-ratings	single-unit bounded op	0.61	0.45	0.174	0.40
Amazon-ratings	bounded message MLP	0.10	0.05	0.135	2.93
Computers	single-unit bounded op	0.95	0.75	0.149	0.20
Computers	bounded message MLP	0.12	0.09	0.136	1.67
Roman-empire	single-unit bounded op	0.55	0.35	0.458	1.35
Roman-empire	bounded message MLP	0.19	0.07	1.674	23.16
Tolokers	single-unit bounded op	0.67	0.34	0.186	0.58
Tolokers	bounded message MLP	0.14	0.06	0.120	2.46
ogbn-arxiv	single-unit bounded op	0.65	0.43	0.159	0.38
ogbn-arxiv	bounded message MLP	0.10	0.04	0.103	2.47

G Pilot disclosure

Four of the five datasets (all but Amazon-ratings) had been used once in an earlier, unrelated evaluation of the branched operator with different code and protocol; Amazon-ratings had never been used. A validation-only pilot (three seeds, four datasets, no test metric) preceded the frozen protocol. Two implementation issues were found and fixed before freezing: (i) an earlier version normalised pre-activations by a batch statistic, which created a train/evaluation mismatch and removed channel-scale freedom (a synthetic magnitude task lost 15 points); the calibrated statistic and the output gain γ replace it; (ii) the tuned dropout value was not propagated into the model in the pilot (all pilot models used dropout 0.5), so all families were re-tuned with the corrected code before the confirmatory batch. The pilot records are kept and are not used for any number in this paper. Soft Median, the message MLP and the branched operators ran out of memory on ogbn-arxiv in the pilot because the GPU was shared; they were run sequentially in the confirmatory batch.

H Full experimental details

Scaffold. Input encoder Linear($d \to 64$) + dropout; two message-passing layers, each followed by a linear residual Linear($\cdot$, no bias); LayerNorm, ELU and dropout between the layers; the second layer maps to the classes. The message MLP uses $H = 256$ hidden units ($4\times$ the width); the single-unit operator $H = 64$; the gated operator $M = 4$ branches of width 64. Multi-head baselines use 4 heads of 16 in layer 1 and one head in layer 2. Parameter counts are in Table 5; the parameter-matched GCN and GATv2 have their hidden width solved so that the total count is within 5% of the message MLP.

Table 5: Trainable parameters.

model	Amazon-ratings	Roman-empire	Tolokers	Computers	ogbn-arxiv
bounded message MLP	74,309	78,482	54,786	105,802	74,536
single-unit bounded op	28,362	30,052	9,412	58,900	21,904
gated 4-branch	42,093	46,370	22,546	73,626	42,600
GCN	28,293	29,970	9,346	58,826	21,800
GCN (param.-matched)	74,625	78,880	54,742	106,328	74,422
GraphSAGE	28,293	29,970	9,346	58,826	21,800
GATv2	32,916	35,464	13,768	63,784	28,768
GATv2 (param.-matched)	73,716	78,136	55,316	104,872	73,960
Soft Median	28,293	29,970	9,346	58,826	21,800
MLP	28,293	29,970	9,346	58,826	21,800

Operating point. Table 6 gives the mean absolute layer-1 pre-activation and the fraction of pre-activations with $|z| > 2\tau$ (saturation rate) at the selected checkpoints, for the calibrated ($\tau = 1$) and the uncalibrated message MLP.

Table 6: Operating point of the calibrated and uncalibrated message MLP (mean over seeds).

dataset	calibrated ($\tau = 1$)			uncalibrated		
	mean $\lvert z\rvert$ L1	sat. L1	sat. L2	mean $\lvert z\rvert$ L1	sat. L1	sat. L2
Amazon-ratings	0.73	4.9%	3.9%	0.014	0.00%	0.1%
Roman-empire	0.67	7.4%	2.5%	0.575	6.47%	54.5%
Tolokers	0.84	3.3%	3.9%	0.206	0.01%	0.1%
Computers	0.67	3.9%	2.3%	0.004	0.00%	0.1%
ogbn-arxiv	0.78	4.8%	3.5%	0.057	0.00%	3.3%

Calibration pseudo-code. Per epoch: (1) model.train(); one full-batch gradient step on the training nodes with dropout, using the stored statistics r (never a batch statistic); (2) model.eval(); calibrating=True; one forward pass on the clean training graph with dropout off, during which each normalisation module overwrites $r_c \leftarrow \text{mean}_j z_{jc}^2$ over all nodes of the graph (transductive setting; labels are not used) layer by layer in forward order, so that the second layer is calibrated on first-layer outputs that already use the fresh first-layer statistic; the same pass produces the validation metric; (3) calibrating=False. The checkpoint stores r with the weights; at test time and under every corruption or attack r is fixed. The initialise-once control performs step (2) once, before training, and never again.

Training and tuning. Adam, full batch, cross-entropy, gradient-norm clipping at 2, up to 500 epochs with patience 100 (50 on ogbn-arxiv), checkpoint at the best validation metric. Learning rate $\{0.001, 0.003, 0.01, 0.03\}$, weight decay $\{0, 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}\}$ and dropout $\{0.2, 0.5, 0.7\}$: the same 12 random combinations of this grid for every family, selected on the validation split of seed 0 (the test mask is physically removed in this mode). Variants inherit the values of their family (Table 7). Scale parameters are excluded from weight decay. A calibration pass (evaluation mode, clean training graph) precedes every epoch's validation pass and sets the RMS statistics; they are stored with the checkpoint and never updated at test time.

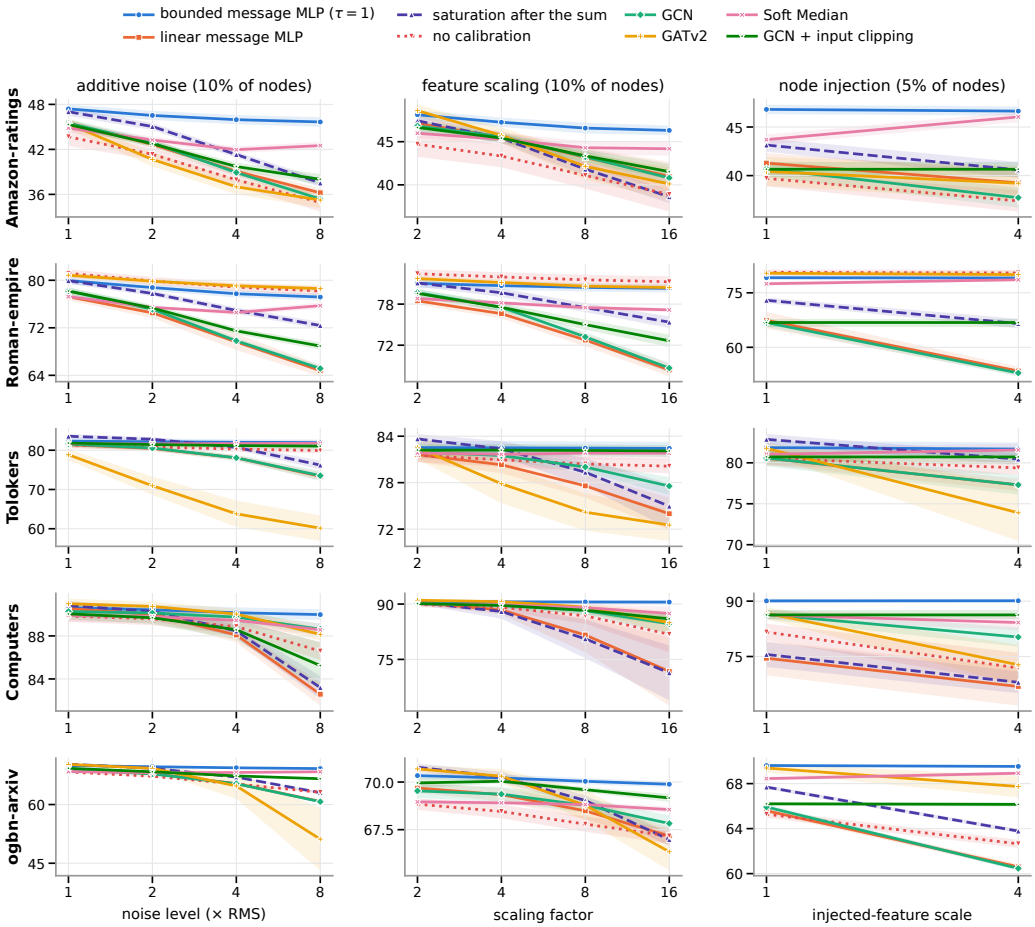

Figure 6: Corrupted test accuracy vs. corruption level (mean ± SD over seeds). Columns: additive noise on 10% of the non-evaluated nodes; feature scaling of 10%; injection of 5% nodes with in-range (×1) or out-of-range (×4) features. Source data and plotting scripts are in the supplementary material.

Corruptions. Masks are drawn with a fixed generator per corruption seed and are identical for every model. Test nodes are never corrupted. Additive noise adds a random unit direction scaled by $\ell\times$ feature RMS $\times\sqrt{d}$; scaling multiplies the feature vector by ℓ; injection adds $\lfloor fn \rceil$ nodes with 10 random edges each and per-dimension features at the training minimum or maximum (times the scale). Five corruption seeds per trained model; levels as in Section 5.

Attacks. PGD with step $2.5\varepsilon/50$, 50 steps, 3 random restarts, on the features of a random 10% of the non-target nodes (shared mask), maximising the cross-entropy of the test nodes; gradient modes: true, straight-through (identity backward through ϕ_τ) and transfer from the linear message model of the same seed; the reported value is the minimum accuracy over modes and restarts. Adaptive injection optimises the injected features inside the per-dimension training range with the same PGD schedule. Attacks were run for the eleven models of the primary comparisons (protocol amendment; the subset was fixed before any attack result was computed).

Compute. One RTX 3090 shared with other users; about 35 GPU-hours for training, 12 for corruptions and 25 for attacks. Every raw record carries the protocol hash, code hash, git commit, data and split hashes, seeds, status and command line. No protocol run failed;

out-of-memory events caused by the shared GPU were retried under identical settings and all attempts are kept in the raw logs.

I Connection-targeted adaptive injection and attack diagnostics

Protocol versions. v1 (pre-registered): random connections, box $[\min \cdot k, \max \cdot k]$, fresh restarts per budget. v2 (revision 3): nested budgets with a warm start, lowest-degree connections with a per-target cap, edge diagnostics stored. v3 (revision 4, all numbers of Section 6): centred boxes $\mathcal{B}_k = [c - kr,\ c + kr]$ with $c = (\min + \max)/2$, $r = (\max - \min)/2$ per coordinate, so $\mathcal{B}_1 = [\min, \max]$ and $\mathcal{B}_1 \subseteq \mathcal{B}_4$ also when a coordinate's range does not contain zero (one coordinate of ogbn-arxiv) and zero-width coordinates stay fixed; a strict cap of two injected neighbours per target with the surplus edges of the fixed budget going to random non-target nodes (in v2 the fallback could hit capped targets: up to five injected neighbours per target on Computers, three on Amazon-ratings); feasibility (cap, box, no duplicate edges, no self-loops, symmetric edges) asserted for every attack graph, an infeasible graph being logged as failed and excluded; and the straight-through surrogate now wraps the saturating map of the message MLP as well (in v1/v2 it wrapped only the single-unit operator, so the "straight-through" numbers of the message-MLP family equalled the true-gradient ones). Capacity: with 5% injected nodes and ten edges each, the cap-two capacity of the test set is exceeded on Computers (6880 edges requested, 5502 available: 20% of the edges go to non-targets) and marginally on Amazon-ratings (12250 vs 12246); Tolokers is exactly at capacity; Roman-empire and ogbn-arxiv are within it. The v2 records are kept and reported as v2 wherever they appear (Appendix O).

Naming. The injection attack of Table 12 is feature-adaptive injection with fixed random connections: 5% new nodes, ten edges each to uniformly random existing nodes, features optimised by PGD inside the data range ($\times 1$) or beyond it ($\times 4$) to maximise the cross-entropy of the test nodes. Here we add a stronger attacker that also chooses where to connect: each injected node links to the ten lowest-degree test nodes that have fewer than two injected neighbours (the nodes for which Proposition 2 gives the weakest protection), and its features are then optimised as before; the node and edge budgets are unchanged. Attacks are evasion attacks on the trained model; every record stores the loss trajectory of the first restart, the final losses of all three restarts, the true-gradient, straight-through and transfer results, the attack success rate (fraction of correctly classified test nodes whose prediction is flipped), the metric on all original nodes, and the metric on the lowest-degree decile of the test nodes.

Convergence and masking. The loss gain over the last 10% of steps is below 1% of the total gain in every record and the three restarts agree within 5%, so the reported worst cases are converged; the straight-through and transfer attacks were never stronger than the true-gradient attack. We treat this as necessary, not sufficient, evidence against gradient masking.

Results (Table 13; v2 records). The targeted attacker is far stronger than the random one for every graph model and separates the families more clearly than any other experiment. On Amazon-Computers the input-normalised operators keep 75–84% attacked accuracy (GCN + input RMSNorm 75.5, GRB scheme 83.2, Soft Median + input RMSNorm 82.4, GraphSAGE + input RMSNorm 78.8) and the bounded family 66–71%, where GCN keeps 17%, GATv2 8%, the post-sum, linear and aggregate-normalised variants 9–11%, input clipping 18% and Soft Median 46%; on Roman-empire the normalised operators keep 47–53% against GCN 4%, GATv2 14% and Soft Median 40%; on Amazon-ratings every graph model is severely damaged (20–31% for the normalised ones, 10–15% for GCN, GATv2 and the linear variants); on Tolokers all models keep 55–72%; on ogbn-arxiv the normalised operators keep 53–56% against GCN 18% and GATv2 10%. Attack success rates (fraction of correctly classified test nodes flipped) are 8–18% for the normalised GCN variants on Computers and 76–92% for GCN and GATv2. The no-graph MLP is unaffected by construction and keeps 41.9, 65.5, 73.1, 84.0 and 53.7%: no graph model exceeds it on four graphs (Soft Median + input RMSNorm comes within 1.6 on Computers) and the best ones tie it on ogbn-arxiv,

so at this attack strength the graph is no longer an asset. With out-of-range (×4) injected features the v2 envelopes are at most 2.7 points below the in-range values for the normalised operators and up to 18 below for the unnormalised ones; the v1 claim that Soft Median is stronger out of range was an artefact of the weaker v1 ×4 attack (Appendix O). Low-degree targets are hit as hard as the rest for every model: two optimised injected neighbours on a node of degree one or two exceed what any bound can absorb, in agreement with the $k/(D+k)$ dependence. The adapted AirGNN (tuned γ) is not robust to this attack (13, 6, 65, 67, 22%).

J A dendritic operator with a structural prior (E5)

Motivation. Dendritic artificial networks (Chavlis & Poirazi, 2025) owe part of their robustness and parameter efficiency to a structural prior: each dendrite samples a fixed, sparse receptive field of the input and each soma reads a fixed set of dendrites. Appendix O showed that branching without such a prior (the gated four-branch operator) adds nothing to the bounded message MLP. Here we ask whether the prior itself is a substitute for the placement rule, or whether it too needs the norm of what enters the neighbour sum to be pinned.

Operator. The gated four-branch operator of Appendix A ($M = 4$ dendrites per output channel, learned static gates) with two changes: (i) a fixed binary mask on W_1, drawn once per run, that lets each dendrite read a random 25% of the input coordinates (the graph analogue of the localised receptive fields of Chavlis & Poirazi (2025); node features have no spatial order, so the subsets are random) and stops the gradient of the masked synapses; (ii) the dendritic nonlinearity, leaky ReLU (slope 0.01) as in Chavlis & Poirazi (2025) or tanh. Five variants at the fixed hyper-parameters of the bounded family, ten seeds (five on ogbn-arxiv), full corruption grid, adaptive attacks and the connection-targeted injection: leaky-ReLU dendrites (unbounded, prior only); tanh dendrites uncalibrated and calibrated; and each nonlinearity with the per-node input RMSNorm of Appendix O. Pre-specified prediction (protocol file, entry E5): the prior alone gives no robustness, the uncalibrated tanh operator stays in its linear regime except on Roman-empire, and the input RMSNorm restores the robustness for both nonlinearities.

Results (Table 14; paired contrasts with 95% intervals in the supplementary material). The prediction held on every graph. (i) The prior alone gives no robustness. The leaky-ReLU prior operator is at least as accurate as GCN clean (49.4 vs. 47.1, 80.8/80.1, 83.0/82.1, 89.3/90.3, 70.9/69.4), but its corruption AUC is within −1.6 to +2.7 points of GCN under every family, its accuracy under random-connection adaptive injection is 24–70% against GCN's 16–71%, and under targeted injection it keeps 15.6, 5.0, 62.3, 20.7 and 19.1% against GCN's 13.3, 4.0, 62.3, 17.3 and 17.9% (success rates 74–94% on four graphs). (ii) An unpinned bounded map helps only where the graph saturates it. The uncalibrated tanh prior operator is as fragile as the leaky-ReLU one on Amazon-ratings, Amazon-Computers and ogbn-arxiv (corruption AUC 42.3, 85.0, 67.5 vs. 42.9, 85.9, 66.5; layer-1 saturation $\leq 0.6\%$), and 3.9–14.4 points more robust on Roman-empire and Tolokers, where its second layer saturates by itself (27% and 11%). (iii) Input RMSNorm restores the robustness for both nonlinearities. Leaky-ReLU prior + input RMSNorm minus leaky-ReLU prior is +2.2 to +16.3 AUC points under every family on every graph and +4.0 to +48.2 points under targeted injection (every 10-seed interval excludes zero; the five-seed ogbn-arxiv contrasts sit at the permutation floor $p = 1/16$); tanh + input RMSNorm minus uncalibrated tanh is +2.7 to +11.1 AUC and +46 targeted on Computers, +2.9 to +6.2 AUC and +6.8 targeted on Amazon-ratings, 0 to +3.1 AUC and +21–29 targeted on Roman-empire and ogbn-arxiv, and within +0.7 on Tolokers, where the unpinned model was already saturated. Calibration does the same for the tanh operator, but is 1.5–10 points below the input-normalised version under targeted injection on every graph. (iv) With the norm pinned, the prior is free. The input-normalised prior operator is within −0.3 to +2.1 AUC points of the input-normalised GCN under every corruption, −0.4 to +1.8 points from it clean, and 1.5–6.6 points below it under targeted injection on three graphs (+0.3 to +1.7 on the other two); against the dense calibrated four-branch operator it is 0–1.1 AUC points more robust and −1.1 to +1.3

points clean while using a quarter of the synapses. The structural prior therefore neither substitutes for the placement rule nor conflicts with it.

K Nearest simple controls around GCN and GraphSAGE (E6)

Requested by an external pre-submission review and pre-specified in the revision protocol (entry E6): at the fixed hyper-parameters of the GCN / GraphSAGE family, ten seeds (five on ogbn-arxiv), full corruption grid, adaptive attacks and connection-targeted injection. Controls: unit-L2 normalisation of the operator input (the position of Algorithm 1, line 7 of Hamilton et al. (2017), but applied to the input of every layer including the first); LayerNorm without affine parameters of the input; a radial clip $h \mapsto h \min(1, r/\|h\|)$ whose radius r is the q-quantile of the clean node norms, calibrated like the RMS statistic and frozen, with q chosen on validation among $\{0.5, 0.75, 0.9, 0.95, 0.99\}$ (three pilot seeds); RMSNorm of the aggregate (the position of MsgNorm, without its rescaling); the original MsgNorm of Li et al. (2023) (aggregate L2-normalised, rescaled by the target's input norm and a learnable scalar); and GraphSAGE with its own output normalisation (normalize=True). Paired contrasts against GCN with input RMSNorm (same seeds; bootstrap intervals and sign-flip p) are in the supplementary material. A first radial-clip batch used one clip module shared by both layers, so its calibrated radius was the second layer's norm quantile ($\approx \sqrt{H}$) and it never clipped first-layer inputs; those runs are kept in the raw records, excluded from every table, and replaced by the per-layer version (selected $q = 0.99$, 0.99, 0.9, 0.95, 0.95 on the five graphs; a fixed-q=0.95 twin gave the same corruption and attack values and shares the selected run where $q = 0.95$, so only the selected variant is tabulated). Reading of Table 15: input LayerNorm equals input RMSNorm; input L2 is within 2.3 AUC; the per-layer radial clip is within 1.7 AUC under random corruptions but 22 points below input RMSNorm under targeted injection on Roman-empire (its radius, the 0.99-quantile of clean norms, is not reached by in-range injected features); RMSNorm of the aggregate is at GCN's level; the original MsgNorm recovers part of the robustness; GraphSAGE's own output normalisation recovers none.

L Attack-strength checks (E7)

Pre-specified in the revision protocol (entry E7) after the external review; every run is a new record with its own strategy or budget label, nothing replaces a pre-registered result.

Radial vs. directional corruption. The scaling and noise corruptions of the main text change feature magnitudes. The directional corruption replaces the feature vector of 10% of the non-evaluated nodes by $\|x\| \,\mathrm{normalise}((1-\lambda)\, x/\|x\| + \lambda d)$ with a random unit vector d and $\lambda \in \{0.25, 0.5, 1\}$, so the norm of every corrupted node is preserved exactly (checked per record). Table 15 (column direct.) gives the mean over λ: every operator loses 0.1–1.6 AUC points on every graph, GATv2 2.5 and 3.5 on Amazon-ratings and Tolokers (1.9 and 1.7 with input RMSNorm). The input-normalised operators are the least affected on every graph, but the differences are within 1 point of GCN. This random re-direction under-states the directional vulnerability: the adaptive fixed-norm direction attack of Appendix O (Table 19) costs GCN 17–39 points and the normalised operators 8–20.

Budget nesting. For feature PGD the ℓ_∞ balls are nested; the reported worst case never increased with ε in any of the 75 (graph, model) series, and the $\times 4$ feature box contains the $\times 1$ box in every dimension on four graphs and in 127 of 128 on ogbn-arxiv.

Budget and objective of feature PGD. On Amazon-Computers and Roman-empire, for GCN, GCN with input clipping, GCN with input RMSNorm, the calibrated and the per-message-normalised message MLP and the post-sum model (ten seeds each), the $\varepsilon = 0.2$ attack was re-run with 200 steps and 10 restarts, with the cross-entropy objective and with a margin objective (mean over targets of the largest wrong logit minus the true logit). Cross-entropy with the larger budget reproduces the 50-step/3-restart worst case within 0.3 points on average (per-seed differences from -0.6 to $+5.6$, the reported envelope being the minimum

over budgets). The margin objective is stronger by 0.1–2.4 points on Roman-empire and for the post-sum model on Computers and weaker by 0.1–0.3 on the GCN variants there; no ordering of the main text changes: on Computers GCN and input clipping stay above the bounded models under feature PGD, on Roman-empire the bounded models stay above GCN.

Connection choice and the per-target cap. The pre-specified attack attaches each injected node to the ten lowest-degree test nodes with at most two injected neighbours per target. Two variants were run at the same node and edge budget for nine models (ten seeds; five on ogbn-arxiv): (i) gradient-guided: targets ranked by $\|\partial\mathcal{L}/\partial x_i\|/\sqrt{d_i+1}$ on the clean graph, features optimised, the score recomputed on the attacked graph and the connections rebuilt, two rounds, best round kept (a TDGIA-style alternating scheme (Zou et al., 2021); not a joint optimisation); (ii) the lowest-degree heuristic with the cap raised to ten. Retained accuracy under (i) is 0–3.5 points below the pre-specified attack on Amazon-ratings, Roman-empire, Computers and ogbn-arxiv and 6–13 below on Tolokers, where the degree heuristic was weakest (GATv2 55 → 42, GCN 62 → 50, input-normalised GCN 66 → 60); under (ii) it is within 0.6 points except on Computers (−3.3 to −4.8 for the bounded models and Soft Median, +0.6 for GCN). No ordering changes: the input-normalised GCN remains the most robust operator on Computers (74.5%) and ogbn-arxiv (53.0%) and within 0.2 of GATv2 with input RMSNorm on Roman-empire (46.4 vs. 46.2; per-message-normalised MLP 52.2). The ogbn-arxiv entry for Soft Median under variant (ii) averages four seeds rather than five.

M Own tuning of the recipe and its combination with Soft Median (E9)

Pre-specified in the revision protocol (entry E9). (i) The input-normalised GCN, GraphSAGE and GATv2 had inherited the learning rate, weight decay and dropout of their plain family; they received their own 12-trial tuning on the validation split of seed 0 (same grid and budget as every family). The selected values coincide with the inherited ones on 9 of 15 (graph, operator) pairs (shown as "=" in Table 16); where they differ, clean accuracy moves by −0.1 to +0.5 points and the corruption AUC by −0.1 to +0.4, except that the tuned GATv2 on Roman-empire gains 0.5 clean and loses 10 points under targeted injection. The recipe therefore needs no tuning of its own. (ii) Soft Median aggregation with the same per-node input RMSNorm (Soft Median's hp, and an own-tuned variant): against Soft Median alone, clean accuracy changes by −0.2 to +1.2 points, the corruption AUC by +0.2 to +2.6, and the targeted-injection accuracy by −0.8 (Tolokers) to +36 (Computers); against the input-normalised GCN the combination is within ±1 AUC under corruption and is 9, 5, 7 and 1 points more robust under in-range targeted injection on Amazon-ratings, Roman-empire, Computers and ogbn-arxiv (−1 on Tolokers), making it the most robust operator under that attack on three graphs (v2 records confirm the ordering with out-of-range features). Cost: that of Soft Median (Appendix Q).

N Operating-point intervention on saved checkpoints (E8)

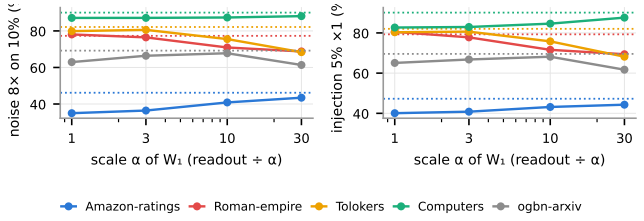

Figure 7: Operating-point intervention on saved uncalibrated checkpoints (three seeds): corrupted metric after multiplying W_1 by α and the readout by $1/\alpha$, no retraining; dotted lines = the calibrated model. Appendix N.

For each of three seeds per graph we take the trained checkpoints of the uncalibrated and the calibrated message MLP and, without retraining, multiply the first-layer projection W_1 by

$\alpha \in \{1, 3, 10, 30\}$ and the readout weight by $1/\alpha$ (for the calibrated model the frozen statistic is divided by α^2, which has the same effect on z). The compensation is exact only where tanh is linear; Table 17 therefore records how much the function changes (clean loss and metric) next to the quantities the bound involves (mean $|z|$, layer-1 saturation, message norm $\|m_j\|$, $\max_c \|R_{\cdot c}\|_1$) and the corrupted metric under 8× noise on 10% of the nodes and 5% in-range injection (three corruption seeds). Reading: for the uncalibrated model on Amazon-ratings, Computers and ogbn-arxiv the clean metric is unchanged up to $\alpha = 10$ (within 0.3 points) while the bound constant shrinks with $\|R\|$, so the same messages sit closer to the bound and the corrupted metric rises towards the calibrated model's value; on Roman-empire and Tolokers, which saturate by themselves, every $\alpha > 1$ costs clean accuracy (82.5 → 72.5 and 81.4 → 76.1 at $\alpha = 10$). The scale is selected on the validation split (largest mean validation corrupted metric among scales whose validation clean metric is within 0.5 points of $\alpha = 1$): $\alpha = 10$ on Amazon-ratings and ogbn-arxiv, 30 on Computers, 3 on Tolokers, 1 on Roman-empire, with test clean accuracy unchanged (≤ 0.3) and test robustness under 8× noise 35.0 → 40.9, 62.9 → 67.8, 87.1 → 88.1, 79.9 → 80.6 (calibrated: 46.2, 69.2, 90.0, 82.1) and under injection 40.1 → 43.2, 65.1 → 68.2, 82.7 → 87.6, 80.3 → 80.6. The compensation is exact only in the linear regime: with $N(z) = z/\sqrt{r+\epsilon}$, dividing r by α^2 multiplies z by α up to a factor $\sqrt{(r+\epsilon)/(r+\alpha^2\epsilon)}$, which is within 0.1% of one for $r \approx 1$, $\epsilon = 10^{-6}$ and $\alpha \leq 30$; for the uncalibrated model W_1 itself is scaled. For the calibrated model any $\alpha > 1$ pushes the pre-activations into saturation (rate 0.2–0.96) and costs accuracy and robustness together. The drift/bound ratio stays below 0.33 throughout.

O Audit batch (E10): nested injection budgets, GRB scheme, MLP, direction attack, end-to-end drift

Pre-specified in the revision protocol (entry E10, 2026-09-17) after a second external audit. Records are versioned as version 2; the version-1 records they supersede are kept.

Nested feature budgets (v2). In v1 the ×4 injection attacks were initialised and stepped inside the ×4 box only, so the ×1 solution, which lies inside the ×4 box on four graphs (on ogbn-arxiv one of 128 feature dimensions has a positive minimum, so the ×4 box does not contain it there; the warm start is clamped), was never a candidate and ×4 could report a weaker attack than ×1 (v1: Roman-empire Soft Median 38.7 vs. 65.3). In v2 the ×1 attack is crafted first, its solution seeds one extra restart of the ×4 attack, and the ×4 result is the worst over its own restarts, the warm-started restart and the ×1 solution; each reported number is one real attack graph. Tables 13 and 18 replace the v1 injection columns of Tables 12 and 13 for every model that was re-run.

Edge budget of the targeted attack. With 5% injected nodes, ten edges each and at most two injected neighbours per test node, the required $0.5N$ edges exceed the $2|\text{test}|$ capacity when the test split is smaller than 25% of the nodes: on Amazon-Computers (20% test) the implementation falls back to uniformly random existing nodes for 16% of the edges once the cap is exhausted, and some targets receive up to five injected neighbours; on Amazon-ratings the excess is three edges; on Roman-empire, Tolokers and ogbn-arxiv the cap is respected exactly. Every v2 record stores the number of injected nodes and edges, the maximum injected neighbours per target and the fraction of fallback edges (column fall. of Table 13); graphs are symmetric and duplicate-free.

Results of the v2 injections (Tables 13, 18). The ×1 numbers reproduce v1 (same procedure); the ×4 envelopes are now at or below ×1 for every model, and the v1 anomalies disappear (Roman-empire Soft Median ×4: 65.3 → 39.5; GATv2 31.0 → 13.5; Computers GCN + clipping 38.5 → 17.3): the warm-started restart is the worst one in most records. No ordering of the main text changes. The MLP is unaffected by construction.

Random-connection injection v2 (Table 18). The ×4 envelopes are within 0.3 points of ×1 for every input-normalised operator and 2–16 points below ×1 for the unnormalised ones (GCN, GATv2, linear and post-sum variants, AirGNN); the no-graph MLP (41.9, 65.5,

73.1, 84.0, 53.7%) is beaten by the input-normalised GCN on Roman-empire, Tolokers, Computers and ogbn-arxiv (+1.6, +0.5, +1.3, +8.6) and not on Amazon-ratings (−8.2); by GRB's scheme by +4.4, +1.6, +3.5, +8.5 and −6.9.

GRB scheme and factor controls (Table 20). The GRB scheme is within 1 AUC point of input RMSNorm under corruption; clean it is +0.6, +1.4, +0.9 points above the every-layer RMSNorm on Amazon-ratings, Roman-empire and Tolokers, −0.7 below on Computers and +0.2 on ogbn-arxiv; under targeted injection it is 2–8 points more robust on Amazon-ratings, Roman-empire and Computers (tie on Tolokers and ogbn-arxiv). Raw-feature LayerNorm alone gives the corruption robustness but not the injection robustness (18.6, 52.5, 14.0, 60.9, 34.7% under targeted injection). LayerNorm before the first layer only is within 0.6 points of LayerNorm before every layer under targeted injection on every graph (19.9, 47.7, 65.5, 74.8, 52.7 vs 20.5, 47.4, 66.0, 75.3, 54.2), so the depth of the normalisation is not what matters in this single-stream setting; affine parameters change clean accuracy and corruption AUC by at most 0.5 but move targeted injection by −5.3 on Computers and +2.1 on Roman-empire, so they are not neutral under attack. RMSNorm applied to both streams reaches the GRB numbers (24.8, 84.1, 53.5, 68.0, 54.8%): the factor that separates the GRB scheme from a per-layer input normalisation is the residual stream. The stream × depth ablation of revision 4 (Table 28) tests the "both streams, first layer only" cell that this batch did not contain.

Fixed-norm adaptive direction attack (Table 19). With the norm of every attacked node preserved exactly, the optimised direction costs GCN 17–39 points, GATv2 19–66, the post-sum model 20–61, and the input-normalised or per-message-normalised operators 3–21 (GCN + input RMSNorm 8, 9, 11, 18, 20 on Computers, Tolokers, ogbn-arxiv, Roman-empire, Amazon-ratings). Random re-direction (Appendix L) under-states the directional vulnerability by an order of magnitude for the unnormalised models; the normalised ones are not immune.

End-to-end drift (Table 21). Under ×16 scaling of 10% of the nodes the layer-1 aggregate drift stays at 10–20% of the Proposition 1 bound for the message MLP and 55–95% for the single-unit operator (as in Appendix F); the end-to-end quantities move together with it: the relative logit drift of the test nodes falls from 0.20–0.50 (GCN) and 0.07–0.63 (GATv2) to 0.04–0.14 with input RMSNorm, and the prediction-flip rate from 10–31% and 4–36% to 1–13%. The bound covers one layer; the flip rates are the measured end-to-end effect.

P Synthetic controls for the boundary

Summary (moved from the main text). A bounded message caps what one neighbour can contribute, so labels carried by the magnitude or the count of a neighbourhood are at risk. On synthetic graphs with such labels (Appendix P) the single-unit bounded operator loses 3–6 points to GCN, but so does the single-unit linear operator (2–5), a width effect; the bounded message MLP is within 1–3 of GCN; the only saturation-specific loss appears at $\tau = 0.25$ (12–14).

Contextual SBM ($n = 3000$, $d = 32$, mean degree 20, homophily 0.8; "regular" = every node has exactly 20 neighbours), three seeds, labels defined by the neighbourhood magnitude or by the number of active neighbours (oracle accuracy $\geq 99.7\%$). Factors: aggregation (symmetric, mean, unnormalised sum), a $\log(1 + \text{degree})$ feature appended to the input, and the degree distribution. Fixed hyper-parameters (hidden 32, dropout 0.3, 300 epochs, best validation).

Information cost of the final recipe (Table 23). On labels carried by neighbourhood magnitude or count the per-node input RMSNorm on the message stream costs GCN 7–15 points (e.g. magnitude, variable degrees: 79.5 → 67.2) and GraphSAGE 8–21 (ranges recomputed from Table 23); the loss is the discarded magnitude, not the bound: the radial clip, which keeps in-range magnitudes, is within 1 point of GCN, and the per-message-normalised message MLP, whose hidden vector still encodes the input norm before the normaliser, is 2–5

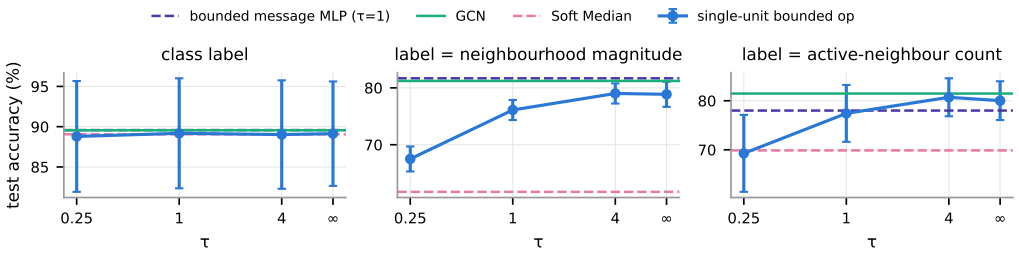

Figure 8: Synthetic boundary: test accuracy of the single-unit bounded operator as a function of τ when the label is the SBM class, the neighbourhood magnitude, or the active-neighbour count (mean ± SD over 3 seeds and 4 graph settings).

points above GCN. The GRB scheme loses most without a degree feature (51–64%) and gains most with one (88–92%, the best of all operators), because its affine LayerNorm can route the degree feature directly. A bounded message is not the same as an inability to count: with sum aggregation the calibrated bounded MLP reaches 57–74% on the count task (Table 22).

Q Cost (E7)

Table 24 reports, per model and graph, the forward-pass time, the calibration-pass time (one clean forward pass in evaluation mode, run once per epoch by the calibrated models; for the others the same pass is timed for reference), the training-step time (forward, backward, optimiser step) and the peak training memory, measured on an idle GPU (NVIDIA, the same device for every row; 50 repeats after 10 warm-up iterations; torch.cuda.synchronize before and after each block; the earlier measurements are kept in the supplementary material). CBMP adds one per-node normalisation per layer and the calibration pass to the message MLP; the message is computed once per sender, never per edge. The calibrated statistic folds into W_1 at inference, so a deployed model pays only the normalisation. Attack-evaluation wall times per checkpoint are in the raw attack records (RUN_MANIFEST.csv).

R Full results of the main comparison and the depth / stream ablation

S Per-run results

Every protocol run completed successfully. Table S lists the clean test metric of every run; per-record corruption and attack values are in the supplementary CSV files.

Table 29: Per-seed clean test metric (%) of every model (protocol seeds).

dataset	model	values by seed
Amazon-ratings	bmp-hardtanh	47.75, 49.50, 48.21, 48.29, 49.08, 48.95, 48.85, 46.56, 49.14, 47.95
Amazon-ratings	bmp-l2clip	47.38, 49.21, 46.66, 47.57, 47.95, 47.53, 48.02, 46.77, 47.30, 47.13
Amazon-ratings	bmp-linear	46.92, 47.71, 46.86, 46.94, 47.95, 47.17, 47.15, 46.58, 47.41, 46.82
Amazon-ratings	bmp-post	47.64, 48.80, 47.75, 47.97, 47.57, 47.51, 47.85, 46.87, 48.15, 46.48
Amazon-ratings	bmp-tanh	48.20, 49.42, 48.05, 48.08, 48.83, 48.33, 49.01, 47.85, 49.11, 47.84
Amazon-ratings	bmp-tanh-nonorm	46.89, 47.57, 47.04, 46.53, 47.64, 47.10, 46.28, 46.66, 47.72, 46.19
Amazon-ratings	bmp-tanh-t0.25	47.77, 48.90, 48.64, 47.92, 47.97, 47.44, 49.01, 47.44, 47.75, 47.95
Amazon-ratings	dgn-M4	47.35, 47.26, 46.64, 46.69, 46.86, 46.45, 47.82, 46.46, 47.30, 46.63
Amazon-ratings	dgn-M4-rms	49.91, 48.41, 48.39, 47.71, 48.95, 49.62, 48.98, 48.15, 47.92, 47.69
Amazon-ratings	gatv2	48.52, 48.70, 48.72, 49.26, 49.98, 50.16, 50.33, 49.21, 48.37, 49.06
Amazon-ratings	gatv2-pm	49.58, 50.22, 50.20, 50.30, 50.02, 49.75, 51.10, 50.65, 49.63, 48.88
Amazon-ratings	gcn	47.75, 47.46, 47.08, 46.50, 47.71, 47.44, 48.23, 46.06, 46.58, 46.30
Amazon-ratings	gcn-inclip	46.63, 48.20, 46.37, 46.50, 47.48, 46.77, 47.84, 46.20, 46.06, 46.95
Amazon-ratings	gcn-pm	47.62, 48.49, 46.06, 46.79, 48.44, 41.73, 47.25, 47.07, 47.74, 47.74
Amazon-ratings	mlp	43.25, 42.64, 42.04, 41.01, 42.61, 39.38, 42.87, 41.70, 41.16, 42.20
Amazon-ratings	mlpmsg	49.24, 49.80, 48.21, 48.23, 48.41, 49.06, 48.31, 48.51, 48.29, 47.85
Amazon-ratings	mlpmsg-hardtanh	48.51, 50.09, 49.26, 48.69, 48.31, 49.13, 49.01, 48.13, 48.26, 47.64
Amazon-ratings	mlpmsg-linear	48.11, 47.28, 47.30, 48.83, 48.20, 47.71, 47.40, 46.81, 47.00, 47.71

Table 29: Per-seed clean test metric (%) of every model (protocol seeds).

dataset	model	values by seed
Amazon-ratings	mlpmsg-nonorm	46.27, 46.48, 45.40, 40.99, 45.71, 46.25, 44.77, 45.24, 45.40, 44.26
Amazon-ratings	mlpmsg-post	48.49, 48.86, 47.74, 47.72, 48.49, 47.79, 47.15, 46.69, 48.08, 48.21
Amazon-ratings	mlpmsg-relu	48.77, 48.44, 48.18, 47.38, 48.44, 49.44, 48.21, 47.80, 47.38, 48.10
Amazon-ratings	mlpmsg-t0.25	48.42, 49.19, 48.08, 48.13, 49.40, 49.18, 48.02, 48.37, 48.41, 47.75
Amazon-ratings	mlpmsg-t0.5	48.33, 48.80, 49.11, 49.08, 48.59, 49.49, 49.11, 49.27, 49.22, 48.15
Amazon-ratings	mlpmsg-t2	48.34, 49.03, 48.95, 47.71, 47.72, 48.98, 48.70, 47.07, 48.26, 47.82
Amazon-ratings	sage	48.60, 48.24, 48.15, 47.67, 48.41, 49.37, 49.39, 48.69, 48.23, 48.47
Amazon-ratings	softmedian	46.76, 47.18, 45.91, 45.99, 46.61, 46.32, 45.86, 45.91, 46.19, 45.09
Computers	bmp-hardtanh	91.57, 91.38, 90.91, 90.77, 90.80, 91.24, 90.95, 90.69, 91.24, 90.00
Computers	bmp-l2clip	91.13, 91.20, 91.17, 90.59, 91.09, 91.53, 90.91, 90.62, 91.24, 89.64
Computers	bmp-linear	91.46, 90.88, 90.80, 90.99, 91.02, 91.64, 90.91, 90.33, 90.99, 89.97
Computers	bmp-post	91.09, 91.24, 91.02, 90.59, 90.84, 91.38, 90.84, 90.51, 91.20, 89.93
Computers	bmp-tanh	91.17, 91.02, 91.31, 90.26, 91.17, 91.28, 90.51, 90.19, 91.42, 90.04
Computers	bmp-tanh-nonorm	89.42, 89.57, 88.95, 88.88, 89.42, 89.17, 89.71, 88.73, 89.02, 88.19
Computers	bmp-tanh-t0.25	91.64, 90.84, 91.28, 90.73, 91.13, 90.99, 91.20, 90.95, 91.57, 90.80
Computers	dgn-M4	89.82, 89.82, 90.08, 89.75, 90.19, 90.00, 90.29, 89.42, 89.89, 88.30
Computers	dgn-M4-rms	91.20, 91.24, 90.84, 90.88, 91.13, 91.06, 90.91, 90.04, 91.35, 90.40
Computers	gatv2	91.68, 91.17, 91.31, 91.24, 91.49, 91.42, 91.20, 90.99, 90.69, 89.93
Computers	gatv2-pm	91.20, 91.06, 91.09, 90.84, 91.17, 91.24, 91.42, 90.84, 90.00, 90.04
Computers	gcn	90.08, 90.22, 90.88, 90.29, 90.77, 90.99, 90.33, 90.44, 90.33, 88.84
Computers	gcn-inclip	89.93, 90.15, 90.22, 90.08, 90.62, 90.66, 90.44, 90.44, 90.29, 88.88
Computers	gcn-pm	90.69, 90.29, 90.69, 89.60, 90.88, 90.51, 90.73, 89.64, 90.29, 89.42
Computers	mlp	83.86, 83.50, 85.02, 83.06, 83.90, 84.81, 85.53, 84.22, 83.64, 82.95
Computers	mlpmsg	91.02, 90.77, 91.09, 91.09, 90.84, 90.99, 91.28, 90.00, 91.06, 89.86
Computers	mlpmsg-hardtanh	91.20, 90.95, 91.28, 91.24, 90.91, 91.28, 90.88, 90.08, 91.38, 90.11
Computers	mlpmsg-linear	90.73, 91.20, 90.66, 91.06, 90.59, 90.80, 90.73, 90.59, 90.84, 89.68
Computers	mlpmsg-nonorm	89.82, 90.29, 90.04, 89.75, 90.33, 90.08, 90.22, 89.82, 90.40, 88.73
Computers	mlpmsg-post	91.17, 91.06, 91.17, 91.46, 91.02, 91.06, 91.17, 90.26, 91.17, 90.29
Computers	mlpmsg-relu	91.09, 91.38, 90.88, 91.35, 91.38, 91.46, 91.57, 90.55, 90.99, 90.19
Computers	mlpmsg-t0.25	91.24, 91.71, 90.99, 91.13, 90.77, 91.06, 91.24, 90.62, 91.31, 90.62
Computers	mlpmsg-t0.5	91.24, 91.53, 91.24, 90.44, 90.91, 91.46, 91.17, 90.37, 91.31, 90.29
Computers	mlpmsg-t2	90.88, 91.02, 91.02, 91.02, 91.17, 90.99, 91.02, 90.08, 91.17, 89.97
Computers	sage	90.55, 91.35, 90.37, 90.80, 90.88, 90.84, 91.17, 90.26, 90.51, 89.49
Computers	softmedian	89.75, 89.24, 90.19, 89.71, 90.69, 90.37, 90.69, 89.86, 89.75, 89.09
Physics	bmp-hardtanh	97.48, 96.87, 96.70, 96.62, 97.17, 97.19, 97.10, 96.97, 96.86, 96.70
Physics	bmp-l2clip	97.43, 96.77, 96.68, 96.46, 97.17, 97.14, 96.94, 96.91, 96.84, 96.71
Physics	bmp-linear	97.32, 96.80, 96.80, 96.49, 97.06, 97.22, 96.96, 96.87, 96.75, 96.72
Physics	bmp-post	97.16, 96.96, 96.58, 96.43, 96.94, 97.09, 97.13, 96.99, 96.80, 96.78
Physics	bmp-tanh	97.49, 97.00, 96.84, 96.65, 97.13, 97.12, 97.16, 96.99, 96.87, 96.80
Physics	bmp-tanh-nonorm	97.23, 96.70, 96.71, 96.90, 97.07, 97.20, 97.07, 97.10, 97.03, 97.09
Physics	bmp-tanh-t0.25	97.52, 97.01, 96.81, 96.88, 97.36, 97.41, 97.29, 97.09, 97.17, 97.07
Physics	dgn-M4	97.29, 96.74, 96.97, 96.96, 97.13, 97.33, 97.14, 97.00, 97.04, 96.96
Physics	dgn-M4-rms	97.20, 96.80, 96.68, 96.61, 96.87, 96.97, 96.97, 96.80, 96.74, 96.75
Physics	gatv2	97.38, 96.90, 96.81, 97.03, 97.36, 97.22, 97.17, 96.91, 97.22, 96.96
Physics	gatv2-pm	97.32, 97.00, 96.87, 97.16, 97.23, 97.20, 97.23, 96.90, 97.06, 97.00
Physics	gcn	97.39, 96.91, 96.96, 97.01, 97.29, 97.17, 97.28, 97.26, 97.25, 97.06
Physics	gcn-inclip	97.42, 96.87, 96.91, 97.12, 97.23, 97.42, 97.13, 97.16, 97.23, 97.12
Physics	gcn-pm	97.41, 96.99, 97.06, 96.94, 97.10, 97.35, 97.28, 97.12, 97.26, 97.22
Physics	mlp	96.48, 95.90, 96.04, 96.10, 96.07, 96.19, 96.28, 96.00, 96.26, 95.93
Physics	mlpmsg	97.10, 96.75, 96.78, 96.48, 96.99, 97.04, 97.04, 96.68, 96.65, 96.84
Physics	mlpmsg-hardtanh	97.14, 96.77, 96.83, 96.42, 97.04, 96.88, 97.04, 96.78, 96.67, 96.81
Physics	mlpmsg-linear	97.23, 96.74, 96.65, 96.46, 97.00, 97.04, 96.86, 96.65, 96.51, 96.67
Physics	mlpmsg-nonorm	97.49, 96.99, 96.94, 97.04, 97.30, 97.49, 97.23, 97.12, 97.29, 96.93
Physics	mlpmsg-post	97.38, 96.77, 96.65, 96.38, 97.04, 97.07, 96.91, 96.70, 96.64, 96.58
Physics	mlpmsg-relu	97.17, 96.94, 96.74, 96.42, 96.96, 97.09, 96.94, 96.71, 96.65, 96.57
Physics	mlpmsg-t0.25	97.49, 96.94, 96.78, 96.48, 97.10, 97.20, 97.14, 96.77, 96.68, 96.87
Physics	mlpmsg-t0.5	97.32, 96.80, 96.70, 96.46, 97.04, 97.07, 97.07, 96.68, 96.78, 96.90
Physics	mlpmsg-t2	97.28, 96.81, 96.70, 96.43, 96.88, 97.06, 96.88, 96.71, 96.64, 96.51
Physics	sage	97.26, 96.80, 96.81, 97.13, 97.10, 97.28, 97.16, 97.03, 97.01, 96.87
Physics	softmedian	97.45, 96.87, 96.90, 97.01, 97.38, 97.29, 97.23, 97.17, 97.03, 97.06
Roman-empire	bmp-hardtanh	78.72, 79.54, 79.63, 79.76, 79.23, 80.83, 79.79, 79.12, 80.52, 78.29
Roman-empire	bmp-l2clip	79.70, 80.85, 79.79, 80.22, 80.46, 81.86, 80.23, 80.13, 80.59, 79.44
Roman-empire	bmp-linear	79.95, 79.76, 79.60, 79.70, 78.82, 80.80, 79.19, 79.99, 79.88, 78.45
Roman-empire	bmp-post	80.20, 80.96, 80.13, 80.27, 79.95, 81.91, 79.99, 80.27, 80.53, 79.58
Roman-empire	bmp-tanh	79.03, 80.85, 79.42, 79.95, 79.37, 81.04, 80.04, 79.60, 79.77, 79.05
Roman-empire	bmp-tanh-nonorm	79.95, 80.53, 79.90, 79.76, 79.77, 81.68, 79.69, 79.93, 79.72, 80.09
Roman-empire	bmp-tanh-t0.25	78.57, 79.47, 78.50, 79.05, 78.26, 79.84, 79.07, 79.24, 78.29, 78.43
Roman-empire	dgn-M4	81.61, 81.87, 81.89, 81.15, 81.27, 82.53, 81.54, 81.29, 80.85, 81.08
Roman-empire	dgn-M4-rms	80.67, 82.30, 80.36, 80.41, 81.01, 81.91, 80.85, 80.69, 80.34, 80.50
Roman-empire	gatv2	81.84, 82.02, 82.03, 81.47, 81.29, 82.46, 81.75, 81.84, 82.32, 81.64
Roman-empire	gatv2-pm	82.67, 82.62, 82.88, 82.55, 81.22, 83.66, 82.53, 82.74, 82.99, 82.99
Roman-empire	gcn	79.77, 80.87, 79.62, 79.62, 79.42, 81.03, 80.23, 80.67, 80.06, 79.47
Roman-empire	gcn-inclip	79.84, 80.94, 79.70, 79.90, 79.47, 81.19, 80.27, 80.50, 79.60, 79.30
Roman-empire	gcn-pm	79.74, 79.74, 79.35, 79.67, 79.63, 80.43, 79.93, 79.63, 79.46, 79.62
Roman-empire	mlp	65.78, 66.10, 65.16, 65.62, 65.18, 66.33, 65.21, 65.11, 65.95, 64.49
Roman-empire	mlpmsg	80.60, 81.66, 81.29, 81.57, 80.59, 82.85, 80.96, 80.71, 81.34, 80.64
Roman-empire	mlpmsg-hardtanh	80.96, 82.51, 80.37, 81.59, 80.97, 82.76, 81.84, 80.69, 81.27, 80.55

Table 29: Per-seed clean test metric (%) of every model (protocol seeds).

dataset	model	values by seed
Roman-empire	mlpmsg-linear	78.47, 80.00, 78.86, 78.56, 78.72, 79.44, 78.56, 78.42, 79.21, 79.02
Roman-empire	mlpmsg-nonorm	82.03, 83.43, 82.09, 82.85, 81.84, 83.78, 82.97, 82.81, 82.30, 82.35
Roman-empire	mlpmsg-post	80.78, 82.30, 80.64, 81.57, 81.19, 82.03, 82.28, 81.45, 81.72, 80.99
Roman-empire	mlpmsg-relu	82.26, 83.34, 82.23, 82.33, 81.66, 84.35, 82.74, 81.98, 80.62, 81.86
Roman-empire	mlpmsg-t0.25	80.96, 81.38, 81.94, 81.72, 80.62, 82.92, 81.96, 81.12, 81.40, 80.96
Roman-empire	mlpmsg-t0.5	81.04, 83.00, 81.29, 81.94, 80.97, 83.73, 81.56, 81.59, 81.98, 81.75
Roman-empire	mlpmsg-t2	80.39, 80.69, 79.40, 80.09, 79.83, 80.99, 80.60, 80.62, 80.94, 79.99
Roman-empire	sage	80.07, 80.55, 79.72, 79.53, 79.16, 80.60, 79.76, 79.92, 79.84, 79.47
Roman-empire	softmedian	79.62, 78.66, 78.20, 78.82, 79.00, 79.33, 79.63, 79.12, 79.21, 78.89
Tolokers	bmp-hardtanh	82.89, 82.94, 83.07, 83.33, 82.71, 81.91, 82.61, 81.90, 83.02, 82.47
Tolokers	bmp-l2clip	81.51, 82.84, 84.48, 83.41, 83.71, 81.26, 82.06, 82.22, 82.54, 82.08
Tolokers	bmp-linear	81.39, 82.02, 83.56, 82.98, 83.89, 82.03, 82.33, 82.63, 82.84, 82.65
Tolokers	bmp-post	82.50, 82.97, 83.39, 82.94, 82.78, 82.40, 83.35, 80.15, 83.65, 83.96
Tolokers	bmp-tanh	82.02, 83.06, 83.55, 83.54, 83.18, 81.23, 82.93, 81.54, 83.10, 82.74
Tolokers	bmp-tanh-nonorm	82.19, 82.30, 83.59, 82.93, 83.77, 81.51, 82.75, 82.42, 83.10, 83.04
Tolokers	bmp-tanh-t0.25	80.35, 82.52, 83.54, 82.72, 83.34, 80.57, 82.10, 81.54, 82.61, 82.41
Tolokers	dgn-M4	82.35, 82.54, 84.24, 83.39, 84.07, 82.11, 83.05, 81.71, 83.55, 82.61
Tolokers	dgn-M4-rms	82.49, 83.02, 83.98, 83.18, 83.50, 82.19, 82.73, 82.37, 83.37, 83.68
Tolokers	gatv2	83.38, 83.72, 85.04, 83.96, 84.33, 82.36, 83.80, 83.77, 83.58, 84.87
Tolokers	gatv2-pm	82.86, 83.03, 83.61, 84.12, 82.67, 82.40, 82.41, 82.41, 82.91, 83.67
Tolokers	gcn	81.25, 81.38, 83.01, 82.13, 82.88, 81.13, 82.53, 80.70, 82.42, 83.14
Tolokers	gcn-inclip	81.01, 81.81, 82.91, 82.39, 82.80, 81.37, 82.60, 81.06, 82.76, 83.09
Tolokers	gcn-pm	81.11, 81.66, 84.20, 83.14, 83.30, 82.18, 82.59, 81.65, 82.14, 82.98
Tolokers	mlp	72.97, 72.80, 74.76, 73.42, 73.84, 71.82, 73.91, 71.32, 72.81, 73.58
Tolokers	mlpmsg	81.92, 82.61, 83.75, 82.85, 83.54, 80.97, 82.82, 81.65, 83.03, 82.97
Tolokers	mlpmsg-hardtanh	81.89, 82.51, 83.75, 82.92, 83.41, 80.93, 82.60, 81.57, 83.26, 82.89
Tolokers	mlpmsg-linear	81.06, 80.95, 82.65, 82.46, 83.06, 80.54, 82.14, 81.18, 82.18, 82.62
Tolokers	mlpmsg-nonorm	80.85, 80.80, 82.58, 82.11, 82.35, 80.46, 82.19, 80.73, 82.21, 82.56
Tolokers	mlpmsg-post	83.42, 83.93, 84.46, 84.37, 84.24, 82.81, 84.01, 83.24, 84.12, 84.47
Tolokers	mlpmsg-relu	82.29, 82.69, 84.53, 83.24, 84.01, 81.55, 82.70, 82.27, 83.10, 83.48
Tolokers	mlpmsg-t0.25	82.65, 83.06, 84.10, 82.97, 83.78, 81.37, 82.77, 82.13, 83.46, 83.36
Tolokers	mlpmsg-t0.5	81.97, 82.92, 84.36, 83.28, 83.63, 81.16, 82.61, 81.84, 83.35, 83.34
Tolokers	mlpmsg-t2	81.63, 81.57, 83.49, 82.76, 83.42, 80.92, 82.66, 81.68, 82.87, 82.78
Tolokers	sage	81.74, 81.88, 83.93, 82.94, 82.40, 81.88, 81.99, 81.63, 81.93, 82.71
Tolokers	softmedian	81.18, 81.55, 83.14, 82.14, 82.78, 80.74, 82.05, 81.25, 82.24, 82.74
ogbn-arxiv	bmp-hardtanh	68.99, 68.93, 68.62, 69.24, 69.16
ogbn-arxiv	bmp-l2clip	69.54, 68.84, 69.27, 69.58, 69.43
ogbn-arxiv	bmp-linear	69.03, 68.83, 68.69, 69.09, 69.62
ogbn-arxiv	bmp-post	69.77, 68.77, 69.31, 69.79, 69.35
ogbn-arxiv	bmp-tanh	69.12, 69.00, 68.70, 68.80, 69.12
ogbn-arxiv	bmp-tanh-nonorm	69.00, 68.83, 68.96, 69.39, 69.25
ogbn-arxiv	bmp-tanh-t0.25	66.65, 67.06, 67.64, 67.05, 67.94
ogbn-arxiv	dgn-M4	69.85, 69.38, 69.71, 69.86, 69.60
ogbn-arxiv	dgn-M4-rms	69.90, 70.18, 70.02, 70.09, 69.73
ogbn-arxiv	gatv2	70.83, 70.93, 70.63, 70.77, 70.53
ogbn-arxiv	gatv2-pm	71.02, 70.63, 70.65, 70.85, 70.70
ogbn-arxiv	gcn	69.56, 70.01, 69.30, 69.09, 69.21
ogbn-arxiv	gcn-inclip	69.59, 70.06, 70.01, 69.35, 70.01
ogbn-arxiv	gcn-pm	70.48, 70.24, 69.46, 69.67, 69.37
ogbn-arxiv	mlp	53.40, 53.79, 53.68, 53.89, 53.51
ogbn-arxiv	mlpmsg	70.36, 70.16, 70.51, 70.21, 70.42
ogbn-arxiv	mlpmsg-hardtanh	70.37, 70.30, 70.44, 70.90, 70.63
ogbn-arxiv	mlpmsg-linear	69.56, 69.78, 69.47, 69.59, 69.88
ogbn-arxiv	mlpmsg-nonorm	68.73, 68.70, 68.71, 69.06, 69.17
ogbn-arxiv	mlpmsg-post	70.96, 70.86, 70.74, 70.54, 71.05
ogbn-arxiv	mlpmsg-relu	70.12, 70.20, 70.21, 70.35, 70.19
ogbn-arxiv	mlpmsg-t0.25	69.50, 69.63, 69.39, 69.47, 69.77
ogbn-arxiv	mlpmsg-t0.5	70.02, 69.95, 69.92, 70.25, 70.38
ogbn-arxiv	mlpmsg-t2	70.10, 69.97, 70.05, 69.98, 70.15
ogbn-arxiv	sage	70.02, 69.85, 70.07, 69.66, 70.25
ogbn-arxiv	softmedian	69.16, 68.71, 68.97, 68.83, 68.95

T Statistics: intervals, Holm-corrected tests and the non-inferiority margin

Three notions are used and kept apart. (1) A paired-bootstrap 95% interval of the mean difference over seeds (10,000 resamples) is the primary evidence for every comparison; an interval that excludes zero is read as a difference, an interval that contains zero is read as "not distinguishable at this sample size", never as equivalence. (2) An exact sign-flip permutation p is computed as a sensitivity check and Holm-corrected within the pre-specified family of a hypothesis; with ten paired seeds its floor is $2/2^{10} \approx 0.002$ and with five it is $1/16$, which is why some ogbn-arxiv differences with intervals far from zero have $p_{\text{Holm}} = 0.56$. (3) Non-inferiority uses the pre-registered margin of 0.5 points: model A is non-inferior to B only if

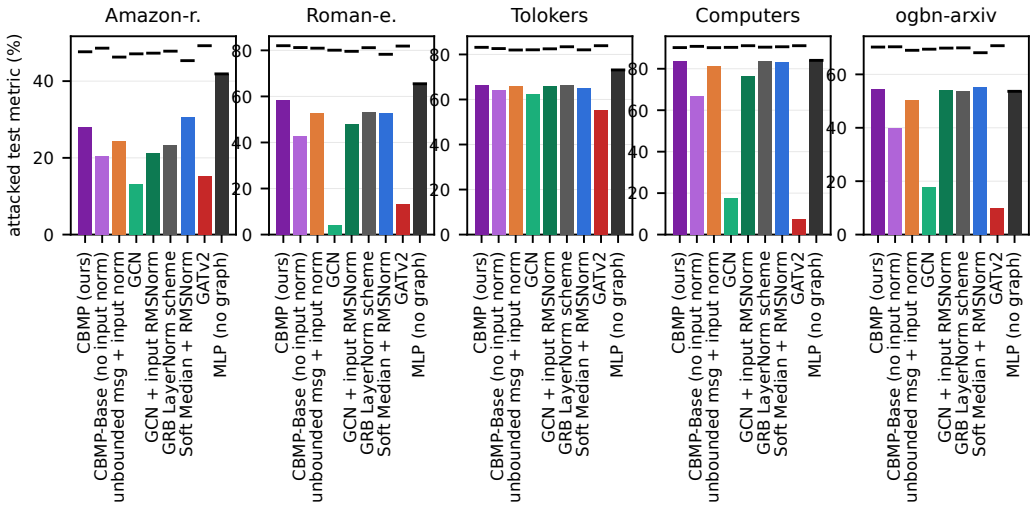

Figure 9: Connection-targeted feature-adaptive injection (v3; 5% nodes linked to the lowest-degree test nodes, cap two; worst case over gradient modes; mean over seeds): attacked accuracy with in-range ($\times 1$, solid) and out-of-range ($\times 4$, light) features; black marks = clean accuracy. Source data and plotting scripts are in the supplement.

the lower interval bound of $A - B$ exceeds -0.5; the bounded message MLP is non-inferior to GCN on all five graphs and to GATv2 on none. The corruption AUC is the unweighted mean of the corrupted metric over the four levels of a family (two for injection: $\times 1$, $\times 4$ at the given fraction), computed per trained model after averaging the five corruption seeds. $\tau^\star$ was selected on validation among $\{0.25, 0.5, 1, 2\}$ and is reported only as a curve; all headline numbers use $\tau = 1$.

Table 7: Selected hyper-parameters per family (validation only).

dataset	family	lr	weight decay	dropout
Amazon-ratings	bmp-tanh	0.01	0.0001	0.2
Amazon-ratings	gcn	0.01	0.0001	0.2
Amazon-ratings	sage	0.01	0.0001	0.2
Amazon-ratings	gatv2	0.01	0.0001	0.2
Amazon-ratings	mlpmsg	0.001	0.0001	0.2
Amazon-ratings	mlp	0.01	0.0001	0.2
Amazon-ratings	softmedian	0.01	0.0001	0.2
Amazon-ratings	airgnn	0.01	0.0001	0.5
Amazon-ratings	mlpmsg-msgrms	0.001	0.0001	0.2
Amazon-ratings	mlpmsg-msgrms-linear	0.001	0.0001	0.2
Amazon-ratings	mlpmsg-meanvar	0.001	0.0001	0.2
Amazon-ratings	gcn-innorm-tuned	0.001	0.0001	0.2
Amazon-ratings	gatv2-innorm-tuned	0.01	0.0001	0.2
Amazon-ratings	sage-innorm-tuned	0.01	0.0001	0.2
Amazon-ratings	softmedian-innorm-tuned	0.001	0.0001	0.2
Computers	bmp-tanh	0.001	0.0001	0.2
Computers	gcn	0.001	0.0	0.5
Computers	sage	0.001	0.0	0.5
Computers	gatv2	0.01	0.0001	0.5
Computers	mlpmsg	0.001	0.0001	0.2
Computers	mlp	0.01	0.0001	0.2
Computers	softmedian	0.01	0.0001	0.5
Computers	airgnn	0.03	0.0001	0.5
Computers	mlpmsg-msgrms	0.001	0.0001	0.5
Computers	mlpmsg-msgrms-linear	0.001	0.0	0.5
Computers	mlpmsg-meanvar	0.01	0.0001	0.2
Computers	gcn-innorm-tuned	0.001	0.0	0.5
Computers	gatv2-innorm-tuned	0.001	0.0	0.5
Computers	sage-innorm-tuned	0.001	0.0	0.5
Computers	softmedian-innorm-tuned	0.001	0.0001	0.2
Physics	bmp-tanh	0.01	0.0001	0.2
Physics	gcn	0.001	0.0001	0.5
Physics	sage	0.001	0.0	0.5
Physics	gatv2	0.001	0.0001	0.2
Physics	mlpmsg	0.001	0.0001	0.5
Physics	mlp	0.001	0.0	0.5
Physics	softmedian	0.001	0.0	0.5
Roman-empire	bmp-tanh	0.03	0.0001	0.2
Roman-empire	gcn	0.01	0.0001	0.2
Roman-empire	sage	0.03	0.0001	0.2
Roman-empire	gatv2	0.03	0.0001	0.5
Roman-empire	mlpmsg	0.03	0.0001	0.2
Roman-empire	mlp	0.03	0.0001	0.5
Roman-empire	softmedian	0.01	0.0001	0.2
Roman-empire	airgnn	0.03	0.0001	0.2
Roman-empire	mlpmsg-msgrms	0.03	0.0001	0.2
Roman-empire	mlpmsg-msgrms-linear	0.03	0.0001	0.2
Roman-empire	mlpmsg-meanvar	0.01	0.0001	0.5
Roman-empire	gcn-innorm-tuned	0.03	0.0001	0.2
Roman-empire	gatv2-innorm-tuned	0.03	0.0001	0.2
Roman-empire	sage-innorm-tuned	0.03	0.0001	0.2
Roman-empire	softmedian-innorm-tuned	0.03	0.0001	0.2
Tolokers	bmp-tanh	0.03	0.0001	0.2
Tolokers	gcn	0.03	0.0001	0.2
Tolokers	sage	0.01	0.0001	0.2
Tolokers	gatv2	0.01	0.0001	0.2
Tolokers	mlpmsg	0.001	0.0001	0.2
Tolokers	mlp	0.01	0.0001	0.2
Tolokers	softmedian	0.01	0.0001	0.2
Tolokers	airgnn	0.03	0.0001	0.2
Tolokers	mlpmsg-msgrms	0.001	0.0001	0.2
Tolokers	mlpmsg-msgrms-linear	0.01	0.0001	0.2
Tolokers	mlpmsg-meanvar	0.01	0.0001	0.2
Tolokers	gcn-innorm-tuned	0.01	0.0001	0.2
Tolokers	gatv2-innorm-tuned	0.01	0.0001	0.2
Tolokers	sage-innorm-tuned	0.01	0.0001	0.2
Tolokers	softmedian-innorm-tuned	0.03	0.0001	0.2
ogbn-arxiv	bmp-tanh	0.01	0.0001	0.2
ogbn-arxiv	gcn	0.01	0.0001	0.2
ogbn-arxiv	sage	0.01	0.0001	0.2
ogbn-arxiv	gatv2	0.01	0.0001	0.2
ogbn-arxiv	mlpmsg	0.001	0.0001	0.2
ogbn-arxiv	mlp	0.01	0.0001	0.2
ogbn-arxiv	softmedian	0.01	0.0001	0.2
ogbn-arxiv	airgnn	0.01	0.0001	0.2
ogbn-arxiv	mlpmsg-msgrms	0.001	0.0001	0.2
ogbn-arxiv	mlpmsg-msgrms-linear	0.001	0.0001	0.2
ogbn-arxiv	mlpmsg-meanvar	0.01	0.0001	0.2
ogbn-arxiv	gcn-innorm-tuned	0.01	0.0001	0.2
ogbn-arxiv	gatv2-innorm-tuned	0.01	0.0001	0.2
ogbn-arxiv	sage-innorm-tuned	0.01	0.0001	0.2
ogbn-arxiv	softmedian-innorm-tuned	0.01	0.0001	0.2

Table 8: E1: normalisation variants of the message MLP at fixed hyper-parameters. Paired difference calibrated RMS (τ=1) − variant in clean accuracy and in corruption AUC averaged over the three families (negative = variant better; * = all three intervals exclude zero; per family in Table 9).

variant	Amazon-ratings		Roman-empire		Tolokers		Computers		ogbn-arxiv	
	clean	corr.	clean	corr.	clean	corr.	clean	corr.	clean	corr.
frozen mean/var (calibrated)	+0.0	+0.4	-0.6	-0.2	+1.5	+1.5*	+2.0	+1.2*	-0.4	+0.6*
RMS calibrated once at init	-0.7	-0.2	-2.1	-1.8*	-0.3	-0.4*	-0.7	+0.2	-0.9	+0.0
per-message RMSNorm	-0.6	-1.6*	-0.6	-0.7*	-0.3	-0.5*	-0.7	-0.5*	-0.9	-0.5*
per-message LayerNorm	-0.6	-1.5*	-0.7	-0.8*	-0.3	-0.6*	-0.7	-0.4*	-0.9	-0.6*
per-message RMSNorm, linear unit	+1.5	+0.6*	+0.0	+0.1	+0.1	+0.0	-0.6	-0.3*	-0.3	+0.1
frozen mean/var, linear unit	+2.8	+9.3*	+1.1	+8.8*	+4.3	+9.1*	+1.8	+14.3*	+0.1	+4.6*
aggregate RMSNorm, tanh (no msg norm)	+3.4	+7.8*	-1.8	-1.0*	+0.4	+1.9*	-0.4	+9.4*	-0.9	+1.7*
per-message RMSNorm, saturation after sum	+0.1	-1.0*	-0.8	-0.5*	-1.6	-1.8*	-0.9	-0.5*	-1.4	-1.0*
no normalisation	+3.5	+6.7*	-1.4	-1.2*	+0.9	+1.7*	+0.9	+6.2*	+1.5	+3.7*
linear message (calibrated RMS)	+1.0	+4.9*	+2.3	+10.7*	+0.7	+3.6*	+0.1	+9.8*	+0.7	+3.9*

Table 9: E1 in full: paired difference calibrated RMS (τ=1) − variant in clean test accuracy and in corruption AUC per family (10% noise, 10% scaling, 5% injection); negative = variant better; * = interval excludes zero.

variant	Amazon-ratings				Roman-empire				Tolokers			Computers				ogbn-arxiv			
	clean	noise	scale	inject	clean	noise	scale	inject	clean	noise	scale	inject	clean	noise	scale	inject	clean	noise	scale	inject
frozen mean/var (calibrated)	+0.0	+0.5*	+0.7*	+0.0	-0.6	-0.3*	+0.0	-0.3	+1.5	+1.6*	+1.8*	+1.2*	+2.0	+2.1*	+0.6*	+0.8*	-0.4	+0.5*	+0.6*	+0.5*
RMS calibrated once at init	-0.7	-0.2	-0.2	-0.2	-2.1	-1.7*	-1.5*	-2.1*	-0.3	-0.4*	-0.3*	-0.4*	-0.7	+0.2*	-0.0	+0.5*	-0.9	+0.0	-0.0	+0.0
per-message RMSNorm	-0.6	-1.6*	-1.3*	-1.8*	-0.6	-0.8*	-0.4*	-1.1*	-0.3	-0.5*	-0.5*	-0.6*	-0.7	-0.5*	-0.4*	-0.7*	-0.9	-0.5*	-0.3*	-0.7*
per-message LayerNorm	-0.6	-1.6*	-1.2*	-1.6*	-0.7	-0.8*	-0.5*	-1.2*	-0.3	-0.6*	-0.5*	-0.6*	-0.7	-0.3*	-0.3*	-0.5*	-0.9	-0.6*	-0.3*	-0.8*
per-message RMSNorm, linear unit	+1.5	+0.5*	+0.7*	+0.6*	+0.0	+0.2	+0.3*	-0.2	+0.1	-0.0	+0.1	-0.0	-0.6	-0.3*	-0.2*	-0.6*	-0.3	+0.1	+0.2	-0.0
frozen mean/var, linear unit	+2.8	+9.0*	+6.9*	+12.0*	+1.1	+4.6*	+6.3*	+15.5*	+4.3	+7.3*	+13.1*	+7.0*	+1.8	+2.9*	+28.3*	+11.7*	+0.1	+4.1*	+2.6*	+7.0*
aggregate RMSNorm, tanh (no msg norm)	+3.4	+7.7*	+6.7*	+9.0*	-1.8	-0.9*	-1.0*	-1.2*	+0.4	+1.6*	+2.1*	+1.9*	-0.4	+3.1*	+6.8*	+18.3*	-0.9	+1.7*	+0.3*	+3.2*
per-message RMSNorm, saturation after sum	+0.1	-1.1*	-0.7*	-1.1*	-0.8	-0.3*	-0.5*	-0.7*	-1.6	-1.9*	-1.6*	-2.0*	-0.9	-0.4*	-0.3*	-0.7*	-1.4	-1.0*	-0.7*	-1.2*
no normalisation	+3.5	+6.9*	+5.1*	+8.2*	-1.4	-1.1*	-1.2*	-1.4*	+0.9	+1.5*	+1.7*	+1.8*	+0.9	+1.6*	+3.7*	+13.3*	+1.5	+3.6*	+2.1*	+5.6*
linear message (calibrated RMS)	+1.0	+5.5*	+2.9*	+6.5*	+2.3	+6.9*	+6.6*	+18.7*	+0.7	+3.8*	+4.1*	+2.8*	+0.1	+2.5*	+7.5*	+19.4*	+0.7	+3.8*	+1.4*	+6.5*

Table 10: Robustness: mean corrupted test metric over the level axis (%) under additive noise (10% of nodes) and injected nodes (5%); all three families in Table 11.

model	Amazon-ratings		Roman-empire		Tolokers		Computers		ogbn-arxiv	
	noise	inject	noise	inject	noise	inject	noise	inject	noise	inject
bounded message MLP (τ=1, calibrated)	46.4	46.7	78.4	79.1	82.1	81.8	90.3	90.1	69.5	69.6
per-message RMSNorm (rev.)	48.0	48.5	79.2	80.2	82.7	82.3	90.8	90.8	70.0	70.2
per-message RMSNorm, linear unit (rev.)	45.8	46.1	78.2	79.3	82.2	81.8	90.5	90.7	69.5	69.6
hardtanh	46.8	47.2	78.6	79.5	82.1	81.8	90.5	90.4	69.8	69.9
saturation after the sum	42.7	41.9	76.2	69.7	80.8	81.7	88.2	71.8	67.4	65.7
no calibration	39.5	38.6	79.5	80.6	80.6	80.0	88.7	76.8	66.0	64.0
linear message MLP	40.9	40.3	71.5	60.4	78.3	78.9	87.8	70.7	65.8	63.1
gated 4-branch (calibrated)	46.2	46.5	77.4	77.9	82.4	82.1	90.5	90.5	69.3	69.3
single unit (τ=1)	46.4	46.8	76.7	77.3	82.1	81.8	90.4	90.1	68.2	68.2
GCN	40.7	39.2	72.1	59.9	78.4	78.9	89.7	83.3	65.7	63.2
GCN (parameter-matched)	40.5	38.7	71.9	59.9	79.3	79.9	89.7	83.6	65.8	62.8
GraphSAGE	41.0	39.4	72.6	58.0	77.0	80.6	89.5	83.7	65.7	65.7
GATv2	39.6	39.8	79.6	80.2	68.4	77.8	90.0	79.7	63.9	68.6
Soft Median	43.1	44.9	75.7	78.0	81.7	81.3	89.4	85.2	68.3	68.7
GCN + input clipping	41.4	40.7	73.5	66.8	81.4	80.7	88.4	86.3	67.9	66.2
MLP (no graph)	41.9	41.9	65.5	65.5	73.1	73.1	84.0	84.0	53.7	53.7
AirGNN (adapted, rev.)	41.9	40.9	73.4	63.3	78.6	78.6	87.1	86.9	67.9	64.0
GCN + input RMSNorm (rev.)	46.0	46.3	77.4	78.8	82.0	81.9	91.0	91.1	69.5	69.6
GraphSAGE + input RMSNorm (rev.)	45.8	46.5	76.8	77.1	81.9	82.2	89.7	89.7	69.4	69.6
GATv2 + input RMSNorm (rev.)	46.7	46.9	79.4	80.2	79.8	81.7	90.9	91.3	70.3	70.2

Table 11: Robustness under all three corruption families (mean corrupted test metric over the level axis, %).

model	Amazon-ratings			Roman-empire			Tolokers			Computers			ogbn-arxiv		
	noise	scale	inject	noise	scale	inject	noise	scale	inject	noise	scale	inject	noise	scale	inject
bounded message MLP (τ=1, calibrated)	46.4	47.1	46.7	78.4	80.6	79.1	82.1	82.5	81.8	90.3	90.6	90.1	69.5	70.1	69.6
per-message RMSNorm (rev.)	48.0	48.4	48.5	79.2	81.0	80.2	82.7	82.9	82.3	90.8	91.0	90.8	70.0	70.4	70.2
per-message RMSNorm, linear unit (rev.)	45.8	46.4	46.1	78.2	80.3	79.3	82.2	82.4	81.8	90.5	90.7	90.7	69.5	69.9	69.6
hardtanh	46.8	47.5	47.2	78.6	80.8	79.5	82.1	82.5	81.8	90.5	90.8	90.4	69.8	70.4	69.9
saturation after the sum	42.7	43.3	41.9	76.2	78.4	69.7	80.8	80.1	81.7	88.2	82.6	71.8	67.4	69.3	65.7
no calibration	39.5	42.0	38.6	79.5	81.8	80.6	80.6	80.8	80.0	88.7	86.9	76.8	66.0	68.1	64.0
linear message MLP	40.9	44.2	40.3	71.5	74.1	60.4	78.3	78.4	78.9	87.8	83.0	70.7	65.8	68.7	63.1
ReLU message MLP	43.5	45.3	42.3	73.8	78.1	65.0	77.6	80.9	78.9	88.1	79.5	73.5	66.1	69.2	63.4
gated 4-branch (calibrated)	46.2	47.7	46.5	77.4	80.1	77.9	82.4	82.9	82.1	90.5	90.8	90.5	69.3	69.9	69.3
gated 4-branch (uncalibrated)	41.6	44.7	41.3	78.2	80.7	78.8	82.3	82.8	81.9	88.9	86.4	78.1	67.9	69.5	66.7
single unit (τ=1)	46.4	47.9	46.8	76.7	79.2	77.3	82.1	82.6	81.8	90.4	90.7	90.1	68.2	68.9	68.2
single unit, norm clipping	45.8	47.1	46.3	77.4	79.7	78.0	82.1	82.4	81.8	90.5	90.8	90.5	68.8	69.3	69.0
GCN	40.7	44.1	39.2	72.1	74.8	59.9	78.4	80.3	78.9	89.7	88.2	83.3	65.7	68.9	63.2
GCN (parameter-matched)	40.5	43.6	38.7	71.9	74.2	59.9	79.3	79.8	79.9	89.7	88.5	83.6	65.8	69.1	62.8
GraphSAGE	41.0	44.0	39.4	72.6	75.2	58.0	77.0	78.4	80.6	89.5	88.6	83.7	65.7	68.4	65.7
GATv2	39.6	44.1	39.8	79.6	81.0	80.2	68.4	76.8	77.8	90.0	88.9	79.7	63.9	69.0	68.6
GATv2 (parameter-matched)	40.0	45.5	39.8	80.2	81.6	81.2	66.9	75.8	76.6	89.4	81.7	68.9	64.1	69.1	68.5
Soft Median	43.1	44.9	44.9	75.7	77.9	78.0	81.7	81.8	81.3	89.4	89.1	85.2	68.3	68.8	68.7
GCN + input clipping	41.4	44.2	40.7	73.5	76.2	66.8	81.4	82.2	80.7	88.4	88.5	86.3	67.9	69.7	66.2
MLP (no graph)	41.9	41.9	41.9	65.5	65.5	65.5	73.1	73.1	73.1	84.0	84.0	84.0	53.7	53.7	53.7
AirGNN (adapted, rev.)	41.9	42.3	40.9	73.4	74.0	63.3	78.6	79.0	78.6	87.1	87.3	86.9	67.9	69.7	64.0
GCN + input RMSNorm (rev.)	46.0	46.5	46.3	77.4	79.4	78.8	82.0	82.3	81.9	91.0	91.1	91.1	69.5	69.8	69.6
GraphSAGE + input RMSNorm (rev.)	45.8	47.0	46.5	76.8	79.0	77.1	81.9	82.0	82.2	89.7	89.9	89.7	69.4	69.8	69.6
GATv2 + input RMSNorm (rev.)	46.7	49.2	46.9	79.4	81.2	80.2	79.8	81.9	81.7	90.9	91.2	91.3	70.3	70.5	70.2

Table 12: Adaptive attacks on test nodes, worst-case test metric (accuracy %; ROC-AUC on Tolokers) over gradient modes, mean over seeds. PGD on the features of 10% of the non-target nodes (ℓ_∞ radius $\varepsilon\times$ feature RMS) and adaptive injection of 5% nodes with optimised in-range features and random connections.

dataset	attack	bounded MLP	msg-RMSNorm	GCN+inRMS	GATv2+inRMS	hardtanh	post-sum	no calib.	linear MLP	4-branch	single unit	GCN	GATv2	Soft Med.	GCN+clip	AirGNN
Amazon-ratings	PGD ε=0.05	42.6	–	–	–	42.6	43.1	39.6	40.9	42.6	41.8	40.8	40.8	39.8	40.7	–
	PGD ε=0.2	32.7	32.6	31.0	24.1	33.1	33.4	29.6	29.7	32.8	32.2	30.3	25.4	30.7	30.3	34.6
	PGD ε=0.5	27.6	–	–	–	28.5	25.3	21.8	22.5	28.0	28.3	22.8	18.2	25.5	22.6	–
	adaptive injection	32.5	36.6	33.7	26.6	33.3	19.4	19.4	16.2	33.2	34.5	16.0	17.0	30.4	15.5	23.6
Roman-empire	PGD ε=0.05	78.0	–	–	–	78.2	77.7	79.2	75.0	77.3	76.1	75.6	79.1	74.8	75.6	–
	PGD ε=0.2	70.5	69.3	66.1	69.5	70.9	69.7	71.1	66.8	69.9	68.9	67.3	71.8	67.0	67.2	68.6
	PGD ε=0.5	65.0	–	–	–	65.4	63.6	65.7	60.6	64.3	64.7	61.3	66.2	62.4	61.3	–
	adaptive injection	65.9	69.9	67.1	65.4	66.5	48.4	66.9	42.3	63.8	65.7	41.0	54.3	65.2	41.1	41.1
Tolokers	PGD ε=0.05	81.8	–	–	–	81.8	83.3	81.1	81.2	82.4	82.0	81.4	83.2	81.2	81.9	–
	PGD ε=0.2	79.5	80.1	79.4	78.8	79.3	81.4	79.1	79.1	80.1	79.8	79.3	80.6	78.9	81.0	78.4
	PGD ε=0.5	75.9	–	–	–	75.8	77.4	75.4	74.4	76.0	76.1	75.2	72.7	75.1	79.3	–
	adaptive injection	72.7	73.3	73.6	65.4	72.8	74.0	71.8	70.3	73.3	73.6	70.6	63.8	73.6	70.7	72.8
Computers	PGD ε=0.05	89.1	–	–	–	89.5	89.5	88.9	89.3	89.4	89.4	89.6	90.0	89.1	89.8	–
	PGD ε=0.2	81.4	83.1	85.1	42.1	81.5	81.4	83.9	78.9	83.1	83.8	86.5	84.7	86.5	88.2	85.1
	PGD ε=0.5	71.8	–	–	–	72.9	56.7	71.5	52.6	74.9	78.9	78.7	66.9	80.4	84.5	–
	adaptive injection	81.5	83.3	85.3	73.7	82.8	57.8	60.3	57.5	83.5	83.6	62.1	51.3	75.3	62.3	82.5
ogbn-arxiv	PGD ε=0.05	68.7	–	–	–	68.7	69.4	67.4	68.1	68.5	67.2	68.0	69.1	67.4	68.3	–
	PGD ε=0.2	63.4	63.1	63.5	57.7	63.5	64.8	62.6	63.0	63.8	62.4	63.4	62.2	63.1	63.7	65.3
	PGD ε=0.5	56.1	–	–	–	57.0	55.8	53.7	53.1	57.3	57.2	54.1	39.0	57.3	54.0	–
	adaptive injection	57.7	60.8	62.3	55.1	58.6	45.2	47.1	43.7	59.1	59.0	43.5	14.8	58.6	43.2	40.0

Table 13: Connection-targeted feature-adaptive injection, v2 (5% nodes, lowest-degree test targets, cap 2; nested budgets with warm start): attacked test metric (accuracy %; ROC-AUC on Tolokers) at $\times$1 and $\times$4 (envelope), attack success rate at $\times$1 (% of test nodes classified correctly before the attack whose argmax prediction flips), and the fraction of edges that fell back to random non-target nodes when the per-target cap was exhausted (fall., %); mean over seeds.

model	Amazon-ratings				Roman-empire				Tolokers				Computers				ogbn-arxiv			
	$\times$1	succ.	$\times$4	fall.	$\times$1	succ.	$\times$4	fall.	$\times$1	succ.	$\times$4	fall.	$\times$1	succ.	$\times$4	fall.	$\times$1	succ.	$\times$4	fall.
MLP (no graph)	41.9	0	41.9	0	65.5	0	65.5	0	73.1	0	73.1	0	84.0	0	84.0	16	53.7	0	53.7	0
GCN	13.3	76	9.9	0	4.0	96	3.4	0	62.3	12	51.1	0	17.3	82	10.4	16	17.9	77	8.5	0
GCN + input RMSNorm	21.4	68	20.5	0	47.9	41	47.2	0	66.0	16	64.2	0	75.5	18	75.1	16	54.0	24	53.2	0
GCN, GRB LayerNorm scheme	23.3	64	23.2	0	53.0	36	52.7	0	66.5	16	66.5	0	83.2	8	83.2	16	53.9	25	53.9	0
GraphSAGE + input RMSNorm	20.6	69	19.6	0	40.1	50	38.9	0	71.9	7	70.8	0	78.8	13	78.0	16	54.7	23	53.9	0
GATv2	15.2	83	13.1	0	13.5	85	13.5	0	55.1	27	44.2	0	7.7	92	3.7	16	10.0	87	6.1	0
GATv2 + input RMSNorm	21.8	73	20.8	0	47.0	44	45.0	0	56.2	29	53.8	0	41.5	55	39.0	16	33.2	55	22.6	0
bounded MLP (calibrated)	20.5	74	20.3	0	42.6	49	40.0	0	64.3	13	58.9	0	65.7	28	64.4	16	40.0	44	37.1	0
bounded MLP (per-msg RMSNorm)	23.9	64	22.9	0	53.2	36	52.6	0	64.8	15	62.7	0	70.8	23	70.7	16	48.1	33	47.9	0
linear MLP	12.0	87	9.3	0	6.0	94	5.1	0	60.1	10	41.9	0	10.8	89	9.0	16	13.8	82	6.6	0
post-sum	14.3	81	12.8	0	12.5	86	7.6	0	67.4	15	51.4	0	8.6	90	7.2	16	19.6	74	9.4	0
aggregate RMSNorm	11.0	88	7.7	0	42.9	49	41.3	0	58.6	11	48.6	0	10.8	89	8.1	16	21.2	71	10.2	0
Soft Median	18.6	74	18.6	0	39.5	51	39.5	0	65.7	8	59.7	0	46.4	49	45.9	16	44.9	36	44.9	0
Soft Median + input RMSNorm	30.6	40	29.7	0	52.8	34	52.4	0	64.9	11	63.6	0	82.4	9	82.3	16	55.3	20	55.3	0
GCN + input clipping	13.2	77	10.7	0	4.1	96	3.9	0	62.4	11	62.4	0	17.7	81	17.3	16	18.2	76	18.2	0
AirGNN (adapted)	12.7	85	8.6	0	6.1	94	4.4	0	65.4	7	49.4	0	66.5	25	22.6	16	22.2	70	3.9	0
single unit	21.5	68	21.4	0	44.0	46	42.6	0	65.1	14	62.3	0	69.7	23	69.1	16	43.8	38	43.4	0
4-branch	21.3	71	20.7	0	36.8	55	32.4	0	64.7	16	61.3	0	69.1	22	67.5	16	45.5	36	42.6	0

Table 14: E5: dendritic operator with a receptive-field prior (fixed hyper-parameters of the bounded family; mean over 10 seeds, 5 on ogbn-arxiv). clean = test accuracy (AUC on Tolokers); corr. = corruption AUC averaged over noise (10%), scaling (10%) and injection (5%); adapt. = worst-case accuracy under feature-adaptive injection with random connections (×1); target. = under connection-targeted injection (×1). Reference rows from the final and rev batches.

operator	Amazon-ratings				Roman-empire				Tolokers				Computers				ogbn-arxiv			
	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.
prior, leaky-ReLU dendrites (unbounded)	49.4	42.9	23.8	15.6	80.8	70.1	39.5	5.0	83.0	79.4	69.5	62.3	89.3	85.9	64.6	20.7	70.9	66.5	45.6	19.1
prior, tanh dendrites, uncalibrated	47.7	42.3	21.6	15.8	81.2	77.7	60.9	30.6	83.2	82.6	74.5	67.2	89.2	85.0	62.8	25.1	70.4	67.5	48.3	22.4
prior, tanh, calibrated	49.4	47.4	32.9	21.1	81.2	78.2	64.5	41.4	83.2	82.8	73.5	65.4	91.6	90.6	82.5	66.8	70.7	69.5	58.8	45.0
prior, leaky-ReLU + input RMSNorm	49.6	47.9	33.8	23.1	79.8	78.5	66.8	46.4	83.7	83.0	73.2	66.4	91.9	91.4	83.7	68.9	71.3	70.1	61.5	50.3
prior, tanh + input RMSNorm	48.2	47.0	34.8	22.6	80.7	79.0	68.5	51.8	83.5	83.1	74.9	67.2	91.6	90.9	83.5	71.2	70.3	69.1	61.0	51.0
dense 4-branch, tanh, uncalibrated	46.9	42.5	–	–	81.5	79.2	–	–	83.0	82.3	73.2	–	89.8	84.5	–	–	69.7	68.0	–	–
dense 4-branch, tanh, calibrated	48.6	46.8	33.2	21.3	80.9	78.5	63.8	36.8	83.1	82.4	73.3	64.7	90.9	90.6	83.5	69.1	70.0	69.5	59.1	45.4
single unit, tanh, calibrated	48.5	47.0	34.5	21.5	79.8	77.7	65.7	44.0	82.7	82.2	73.6	65.1	90.8	90.4	83.6	69.7	68.9	68.4	59.0	43.8
GCN	47.1	41.3	16.0	12.8	80.1	68.9	41.0	3.8	82.1	79.2	70.6	58.1	90.3	87.1	62.1	15.7	69.4	65.9	43.5	17.4
GCN + input RMSNorm	47.8	46.3	33.7	20.4	80.2	78.5	67.1	47.4	82.6	82.1	73.6	63.9	91.7	91.0	85.3	73.7	70.7	69.6	62.3	53.7

Table 15: E6: nearest simple controls (mean over seeds; clean = test accuracy or ROC-AUC on Tolokers; corr. = corruption AUC averaged over noise 10%, scaling 10%, injection 5%; direct. = equal-norm directional corruption AUC (E7, 10%); adapt. = feature-adaptive injection with random connections, ×1; target. = connection-targeted injection, ×1).

operator	Amazon-ratings					Roman-empire					Tolokers					Computers					ogbn-arxiv				
	clean	corr.	direct.	adapt.	target.	clean	corr.	direct.	adapt.	target.	clean	corr.	direct.	adapt.	target.	clean	corr.	direct.	adapt.	target.	clean	corr.	direct.	adapt.	target.
GCN	47.1	41.3	46.0	16.0	13.3	80.1	68.9	78.6	41.0	4.0	82.1	79.2	81.6	70.6	62.3	90.3	87.1	90.2	62.1	17.3	69.4	65.9	68.9	43.5	17.9
GCN + input RMSNorm	47.8	46.3	46.6	33.7	21.4	80.2	78.5	78.3	67.1	47.9	82.6	82.1	82.3	73.6	66.0	91.7	91.0	91.0	85.3	75.5	70.7	69.6	69.6	62.3	54.0
GCN + input L2	47.2	45.8	45.9	35.6	24.7	78.7	77.2	76.8	67.5	51.4	81.7	81.3	81.4	74.3	67.1	91.6	91.0	91.0	85.5	75.3	68.8	67.5	67.3	59.3	46.7
GCN + input LayerNorm	47.8	46.3	46.7	33.6	20.5	80.2	78.5	78.2	67.0	47.4	82.7	82.2	82.4	73.7	66.0	91.7	91.1	91.0	85.4	75.3	70.7	69.6	69.5	62.3	54.2
GCN + radial clip (q on validation)	47.0	45.1	45.5	31.1	19.9	80.8	77.4	78.8	59.6	25.6	82.4	81.8	82.0	72.7	65.2	90.9	89.9	89.9	85.0	77.8	70.6	69.4	69.3	60.3	50.3
GCN + RMSNorm of the aggregate	47.4	40.4	45.6	19.0	10.6	80.7	71.4	78.3	47.2	5.3	82.1	80.0	81.7	70.3	62.0	91.7	85.1	90.9	58.8	15.3	70.9	66.3	69.8	42.9	9.8
GCN + MsgNorm (original)	47.0	41.8	45.4	16.3	10.1	81.0	76.6	79.0	58.2	32.8	81.6	80.0	81.4	71.0	60.3	91.6	90.1	90.9	75.2	53.6	71.3	67.2	70.0	45.8	21.7
GraphSAGE	48.5	41.5	–	–	–	79.9	68.6	–	–	–	82.3	78.7	–	–	–	90.6	87.3	–	–	–	70.0	66.6	–	–	–
GraphSAGE + input RMSNorm	48.1	46.4	46.7	33.4	20.6	79.7	77.6	77.8	62.6	40.1	82.5	82.0	82.2	76.2	71.9	90.9	89.8	89.9	85.0	78.8	70.5	69.6	69.7	62.4	54.7
GraphSAGE, normalize=True	47.9	41.4	46.2	26.0	9.0	78.0	72.8	75.8	58.8	36.6	81.8	78.5	80.5	72.3	65.3	91.6	85.4	90.6	61.8	16.5	68.8	64.0	67.1	50.7	20.3
bounded message MLP (calibrated)	48.6	46.7	47.9	32.5	20.5	81.2	79.4	80.0	65.9	42.6	82.6	82.1	82.3	72.7	64.3	90.8	90.3	90.6	81.5	65.7	70.3	69.7	69.9	57.7	40.0
bounded message MLP (per-message RMSNorm)	49.1	48.3	48.4	36.6	23.9	81.8	80.1	80.1	69.9	53.2	82.9	82.6	82.7	73.3	64.8	91.5	90.8	90.9	83.3	70.8	71.2	70.2	70.2	60.8	48.1

Table 16: E9: own tuning of the input-normalised operators and Soft Median + input RMSNorm (mean over 10 seeds, 5 on ogbn-arxiv; columns as in Table 15; "=" = the tuned hyper-parameters coincide with the inherited ones, same runs).

operator	Amazon-ratings				Roman-empire				Tolokers				Computers				ogbn-arxiv			
	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.	clean	corr.	adapt.	target.
GCN	47.1	41.3	16.0	13.3	80.1	68.9	41.0	4.0	82.1	79.2	70.6	62.3	90.3	87.1	62.1	17.3	69.4	65.9	43.5	17.9
GCN + input RMSNorm (GCN hp)	47.8	46.3	33.7	21.4	80.2	78.5	67.1	47.9	82.6	82.1	73.6	66.0	91.7	91.0	85.3	75.5	70.7	69.6	62.3	54.0
GCN + input RMSNorm (own tuning)	47.7	46.6	34.5	21.8	80.4	78.8	67.2	48.4	82.8	82.4	73.6	65.3	=	=	=	=	=	=	=	=
GraphSAGE	48.5	41.5	–	–	79.9	68.6	–	–	82.3	78.7	–	–	90.6	87.3	–	–	70.0	66.6	–	–
GraphSAGE + input RMSNorm (SAGE hp)	48.1	46.4	33.4	20.6	79.7	77.6	62.6	40.1	82.5	82.0	76.2	71.9	90.9	89.8	85.0	78.8	70.5	69.6	62.4	54.7
GraphSAGE + input RMSNorm (own tuning)	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
GATv2	49.2	41.2	17.0	15.3	81.9	80.3	54.3	13.4	83.9	74.3	63.8	55.1	91.1	86.2	51.3	7.6	70.7	67.1	14.8	10.1
GATv2 + input RMSNorm (GATv2 hp)	50.4	47.6	26.6	21.8	81.8	80.3	65.4	47.0	83.5	81.1	65.4	56.2	92.3	91.1	73.7	41.5	71.5	70.3	55.1	33.2
GATv2 + input RMSNorm (own tuning)	=	=	=	=	82.3	80.3	63.4	36.6	=	=	=	=	92.3	91.5	77.0	44.7	=	=	=	=
Soft Median	46.2	44.3	30.4	18.6	79.0	77.2	65.2	39.4	82.0	81.6	73.6	65.7	89.9	87.9	75.3	46.4	68.9	68.6	58.6	44.9
Soft Median + input RMSNorm (SM hp)	46.0	44.9	37.7	30.6	79.1	77.4	68.6	52.8	82.1	81.9	74.7	64.9	91.1	90.5	88.0	82.4	–	68.0	62.5	55.3
Soft Median + input RMSNorm (own tuning)	47.2	46.5	37.4	27.3	78.6	77.1	68.9	54.6	81.4	81.2	75.1	66.5	91.4	90.8	86.6	77.7	69.1	68.0	62.4	55.3

Table 17: E8: intervention on the operating point of saved checkpoints (mean over three seeds; clean and corrupted values in %; $\|m\|$ = mean message norm).

dataset	model	α	mean $\|z\|$	sat. L1	$\|m\|$	$\max_c \|R_{\cdot c}\|_1$	clean	noise 8×	inject ×1
Amazon-ratings	uncalibrated	1	0.013	0.0%	0.32	16.35	46.1	35.0	40.1
		3	0.039	0.0%	0.32	5.45	46.0	36.4	40.8
		10	0.129	0.0%	0.30	1.63	46.1	40.9	43.2
		30	0.388	0.7%	0.25	0.54	45.4	43.5	44.3
	calibrated	1	0.732	4.9%	1.56	9.45	49.1	46.2	47.3
		3	2.180	42.3%	0.87	3.15	46.3	45.0	45.4
		10	6.759	79.0%	0.32	0.94	39.0	38.5	38.6
		30	13.781	89.5%	0.11	0.31	35.4	35.0	35.2
Roman-empire	uncalibrated	1	0.550	5.8%	9.50	30.42	82.5	78.1	80.6
		3	1.649	24.0%	4.68	10.14	79.4	76.5	77.8
		10	5.496	54.2%	1.71	3.04	72.5	70.9	71.6
		30	16.487	78.7%	0.61	1.01	70.0	68.8	69.4
	calibrated	1	0.665	7.5%	15.87	36.90	81.2	77.3	79.3
		3	1.994	32.4%	8.34	12.30	79.2	76.8	77.6
		10	6.648	71.4%	2.95	3.69	74.4	72.8	73.1
		30	19.936	89.9%	1.02	1.23	72.1	70.8	71.2
Tolokers	uncalibrated	1	0.204	0.0%	8.23	20.34	81.4	79.9	80.3
		3	0.613	1.0%	6.24	6.78	81.4	80.6	80.6
		10	2.044	43.7%	2.74	2.03	76.1	75.5	75.8
		30	6.133	80.2%	0.98	0.68	68.2	68.4	68.2
	calibrated	1	0.838	3.3%	11.84	15.52	82.8	82.1	82.0
		3	2.514	58.7%	5.37	5.17	79.7	79.1	79.4
		10	8.368	87.7%	1.71	1.55	69.1	68.8	69.1
		30	24.817	95.9%	0.57	0.52	62.6	62.8	62.6
Computers	uncalibrated	1	0.004	0.0%	0.11	11.18	90.1	87.1	82.7
		3	0.011	0.0%	0.11	3.73	90.0	87.1	83.0
		10	0.036	0.0%	0.11	1.12	90.0	87.3	84.6
		30	0.109	0.0%	0.11	0.37	90.0	88.1	87.6
	calibrated	1	0.663	3.7%	3.59	8.82	91.0	90.0	90.1
		3	1.260	19.9%	1.55	2.94	90.8	90.2	90.3
		10	1.540	26.9%	0.49	0.88	89.5	88.5	89.0
		30	1.578	27.7%	0.16	0.29	75.6	72.7	73.1
ogbn-arxiv	uncalibrated	1	0.057	0.0%	1.65	16.68	68.7	62.9	65.1
		3	0.171	0.0%	1.57	5.56	68.8	66.4	66.8
		10	0.568	1.5%	1.17	1.67	69.0	67.8	68.2
		30	1.705	32.9%	0.57	0.56	63.5	61.4	61.7
	calibrated	1	0.778	4.8%	2.51	10.12	70.3	69.2	69.6
		3	2.333	48.0%	1.24	3.37	67.3	65.6	66.0
		10	7.686	82.9%	0.43	1.01	24.6	22.3	22.7
		30	21.048	93.7%	0.15	0.34	4.0	4.4	4.0

Table 18: Random-connection adaptive injection v2 (5% nodes, ten random edges; nested budgets): attacked test metric (%).

model	Amazon-ratings		Roman-empire		Tolokers		Computers		ogbn-arxiv	
	×1	×4	×1	×4	×1	×4	×1	×4	×1	×4
MLP (no graph)	41.9	41.9	65.5	65.5	73.1	73.1	84.0	84.0	53.7	53.7
GCN	15.4	12.4	41.0	38.8	70.6	58.0	62.1	56.3	43.5	35.0
GCN + input RMSNorm	33.7	33.6	67.1	67.0	73.6	72.8	85.3	85.2	62.3	62.2
GCN, GRB LayerNorm scheme	35.0	35.0	69.9	69.8	74.7	74.7	87.5	87.4	62.2	62.2
GraphSAGE + input RMSNorm	33.4	33.2	62.6	62.3	76.2	75.7	85.0	84.9	62.4	62.2
GATv2	17.1	17.0	54.3	54.3	63.8	47.3	51.3	35.4	14.8	8.9
GATv2 + input RMSNorm	26.5	25.4	65.4	64.6	65.4	64.8	73.7	72.7	55.1	52.6
bounded MLP (calibrated)	32.5	32.5	65.9	65.2	72.7	69.7	81.5	80.9	57.7	56.7
bounded MLP (per-msg RMSNorm)	36.6	36.2	69.9	69.6	73.3	72.5	83.3	83.3	60.8	60.8
linear MLP	16.2	12.7	42.3	40.5	70.3	54.4	58.1	56.7	43.7	36.2
post-sum	19.4	15.9	48.4	44.4	74.0	61.4	57.7	56.6	45.2	37.2
aggregate RMSNorm	–	–	–	–	–	–	–	–	–	–
Soft Median	30.4	30.4	65.2	65.2	73.6	70.7	75.3	75.2	58.6	58.6
Soft Median + input RMSNorm	37.7	37.5	68.6	68.4	74.7	74.4	88.0	88.0	62.5	62.5
GCN + input clipping	14.9	13.9	41.0	40.7	70.7	70.7	62.3	62.0	43.2	43.0
AirGNN (adapted)	23.6	14.5	41.1	20.1	72.8	61.5	82.5	60.0	40.0	14.5
single unit	–	–	–	–	–	–	–	–	–	–
4-branch	–	–	–	–	–	–	–	–	–	–

Table 19: Fixed-norm adaptive direction attack (10% of the non-target nodes; unit direction optimised on the sphere, norm preserved exactly, no data-range box): clean and attacked test metric (%).

model	Amazon-ratings		Roman-empire		Tolokers		Computers		ogbn-arxiv	
	clean	attacked	clean	attacked	clean	attacked	clean	attacked	clean	attacked
MLP (no graph)	41.9	41.9	65.5	65.5	73.1	73.1	84.0	84.0	53.7	53.7
GCN	47.1	15.1	80.1	57.8	82.1	64.9	90.3	54.6	69.4	30.8
GCN + input RMSNorm	47.3	27.3	79.6	62.0	82.5	73.5	91.1	82.7	69.8	59.0
GCN, GRB LayerNorm scheme	47.8	27.1	81.2	64.4	83.4	74.1	90.4	86.7	69.9	58.0
GraphSAGE + input RMSNorm	–	–	–	–	–	–	–	–	–	–
GATv2	49.2	13.3	81.9	62.6	83.9	48.6	91.1	39.8	70.7	4.4
GATv2 + input RMSNorm	49.6	17.0	81.4	63.3	83.3	56.2	91.8	20.6	70.8	16.2
bounded MLP (calibrated)	48.6	25.1	81.2	63.0	82.6	70.8	90.8	68.4	70.3	49.2
bounded MLP (per-msg RMSNorm)	49.0	29.8	81.3	64.8	83.0	74.6	91.1	78.4	70.5	55.3
linear MLP	–	–	–	–	–	–	–	–	–	–
post-sum	47.9	17.6	81.5	60.3	83.9	63.7	91.0	29.8	70.8	31.4
aggregate RMSNorm	–	–	–	–	–	–	–	–	–	–
Soft Median	46.2	22.4	79.0	60.9	82.0	70.3	89.9	65.5	68.9	53.5
Soft Median + input RMSNorm	45.3	30.3	78.3	62.4	82.0	74.5	90.6	87.8	68.1	60.2
GCN + input clipping	46.9	15.0	80.1	57.8	82.2	74.7	90.2	67.4	69.8	29.8
AirGNN (adapted)	42.5	25.6	78.8	59.4	80.1	66.7	87.2	84.4	70.3	35.0
single unit	–	–	–	–	–	–	–	–	–	–
4-branch	–	–	–	–	–	–	–	–	–	–

Table 20: GRB LayerNorm scheme and factor controls (GCN hp; clean, corruption AUC, directional corruption AUC, v2 random-connection injection ×1, v2 targeted injection ×1, fixed-norm direction attack).

variant	Amazon-ratings						Roman-empire						Tolokers						Computers						ogbn-arxiv					
	clean	corr.	direct.	inj. ×1	targ. ×1	dir. att.	clean	corr.	direct.	inj. ×1	targ. ×1	dir. att.	clean	corr.	direct.	inj. ×1	targ. ×1	dir. att.	clean	corr.	direct.	inj. ×1	targ. ×1	dir. att.	clean	corr.	direct.	inj. ×1	targ. ×1	dir. att.
GCN (no extra normalisation)	47.1	41.3	46.0	15.4	13.3	15.1	80.1	68.9	78.6	41.0	4.0	57.8	82.1	79.2	81.6	70.6	62.3	64.9	90.3	87.1	90.2	62.1	17.3	54.6	69.4	65.9	68.9	43.5	17.9	30.8
raw-feature LayerNorm (affine), before the encoder	48.9	47.1	46.9	–	18.6	–	80.9	79.1	79.0	–	14.0	–	82.9	82.2	82.2	–	60.9	–	91.0	90.2	90.2	–	52.5	–	70.9	69.4	69.3	–	34.7	–
LayerNorm before the first layer only	47.6	46.0	46.5	–	19.9	–	80.5	78.7	78.6	–	47.7	–	82.4	82.0	82.1	–	65.5	–	91.8	91.1	91.1	–	74.8	–	70.7	69.6	69.6	–	52.7	–
LayerNorm before every layer (no affine)	47.8	46.3	46.7	33.6	20.5	–	80.2	78.5	78.2	67.0	47.4	–	82.7	82.2	82.4	73.7	66.0	–	91.7	91.1	91.0	85.4	75.3	–	70.7	69.6	69.5	62.3	54.2	–
LayerNorm before every layer (affine)	47.8	46.2	46.6	–	20.5	–	80.7	79.1	78.9	–	49.5	–	82.6	82.2	82.4	–	65.7	–	91.7	90.9	91.0	–	70.0	–	70.8	69.8	69.7	–	53.5	–
RMSNorm before every layer (no affine; ours)	47.8	46.3	46.6	33.7	21.4	27.3	80.2	78.5	78.3	67.1	47.9	62.0	82.6	82.1	82.3	73.6	66.0	73.5	91.7	91.0	91.0	85.3	75.5	82.7	70.7	69.6	69.6	62.3	54.0	59.0
RMSNorm before every layer, both streams	47.4	46.1	46.1	–	24.8	–	81.1	79.4	79.2	–	53.5	–	82.9	82.6	82.6	–	68.0	–	91.0	90.2	90.2	–	84.1	–	70.4	69.5	69.4	–	54.8	–
GRB scheme: raw + every layer, affine, single stream	48.4	47.2	47.0	35.0	23.3	27.1	81.6	80.1	79.9	69.9	53.0	64.4	83.5	83.0	83.0	74.7	66.5	74.1	91.0	90.3	90.3	87.5	83.2	86.7	70.9	69.6	69.6	62.2	53.9	58.0
RMSNorm of the aggregate	47.4	40.4	45.6	–	–	–	80.7	71.4	78.3	–	–	–	82.1	80.0	81.7	–	–	–	91.7	85.1	90.9	–	–	–	70.9	66.3	69.8	–	–	–
MagNorm (original)	47.0	41.8	45.4	–	–	–	81.0	76.6	79.0	–	–	–	81.6	80.0	81.4	–	–	–	91.6	90.1	90.9	–	–	–	71.3	67.2	70.0	–	–	–

Table 21: End-to-end drift under feature scaling (×16 on 10% of the non-test nodes; three seeds): layer-1 aggregate drift over the Proposition 1 bound (max over test nodes; bounded operators only), relative logit drift of the test nodes, and prediction-flip rate (%).

model	Amazon-ratings			Roman-empire			Tolokers			Computers			ogbn-arxiv		
	L1 drift/bound	logit drift	flips	L1 drift/bound	logit drift	flips	L1 drift/bound	logit drift	flips	L1 drift/bound	logit drift	flips	L1 drift/bound	logit drift	flips
GCN	–	0.50	30.9	–	0.22	18.4	–	0.37	9.9	–	0.42	10.4	–	0.20	11.8
GCN + input RMSNorm	–	0.14	12.6	–	0.04	2.2	–	0.07	2.5	–	0.05	0.8	–	0.08	3.2
GATv2	–	0.59	36.0	–	0.07	4.3	–	0.63	29.2	–	0.38	9.0	–	0.34	16.4
GATv2 + input RMSNorm	–	0.14	9.3	–	0.05	2.0	–	0.14	2.7	–	0.09	2.0	–	0.14	5.9
linear MLP	–	0.41	30.0	–	0.20	17.9	–	0.58	42.9	–	0.58	26.0	–	0.25	13.4
bounded MLP, uncalibrated	0.20	0.48	30.6	0.15	0.07	4.1	0.33	0.17	11.8	0.06	0.38	9.5	0.27	0.17	9.4
bounded MLP, calibrated	0.10	0.22	15.9	0.18	0.05	2.9	0.14	0.05	1.6	0.13	0.10	1.6	0.10	0.09	3.7
bounded MLP, per-msg RMSNorm	0.13	0.12	7.0	0.09	0.03	1.9	0.07	0.05	1.8	0.10	0.06	0.9	0.07	0.07	2.8
single unit	0.58	0.13	10.2	0.55	0.06	3.5	0.64	0.09	2.9	0.95	0.10	1.2	0.62	0.05	2.4

Table 22: Test accuracy (%) on the synthetic magnitude and count tasks (mean over 3 seeds).

task	degrees	degree feat.	aggregation	single-unit tanh	single-unit linear	bounded MLP	GCN
count	regular	no	mean	70.4	71.4	74.2	–
count	regular	no	sum	63.7	61.3	57.0	–
count	regular	no	sym	69.8	73.5	74.3	75.7
count	regular	yes	mean	70.3	66.1	79.1	–
count	regular	yes	sum	62.6	62.4	59.2	–
count	regular	yes	sym	68.2	67.7	81.5	75.5
count	variable	no	mean	72.3	71.0	74.7	–
count	variable	no	sum	71.5	68.2	63.6	–
count	variable	no	sym	71.8	77.0	76.4	78.0
count	variable	yes	mean	75.4	73.2	76.6	–
count	variable	yes	sum	71.4	68.6	66.9	–
count	variable	yes	sym	73.3	75.2	78.9	76.5
magnitude	regular	no	mean	75.5	72.3	76.8	–
magnitude	regular	no	sum	59.0	60.1	55.1	–
magnitude	regular	no	sym	74.6	73.7	75.8	77.6
magnitude	regular	yes	mean	73.4	67.0	83.6	–
magnitude	regular	yes	sum	56.5	56.2	51.0	–
magnitude	regular	yes	sym	73.1	68.4	84.5	78.1
magnitude	variable	no	mean	76.0	76.5	82.4	–
magnitude	variable	no	sum	67.7	62.5	57.9	–
magnitude	variable	no	sym	74.9	76.8	81.7	79.5
magnitude	variable	yes	mean	77.0	74.4	84.8	–
magnitude	variable	yes	sum	66.6	62.8	58.4	–
magnitude	variable	yes	sym	74.2	74.0	82.8	79.4

Table 23: Information cost of the final recipe on the synthetic magnitude / count tasks (symmetric aggregation, mean over 3 seeds; "deg." = a $\log(1 + \text{degree})$ input feature).

operator	magnitude						count					
	regular			variable			regular			variable		
	raw	const	deg.	raw	const	deg.	raw	const	deg.	raw	const	deg.
CBMP	73.9	78.1	86.1	83.9	86.8	88.7	73.2	74.0	84.4	77.7	80.7	82.4
CBMP + budget	71.5	75.9	80.6	80.6	85.2	85.7	67.7	74.2	82.8	75.8	81.9	85.3
unbounded message + input norm	78.5	82.4	90.0	82.6	86.1	87.7	76.8	76.7	85.4	82.2	82.9	86.1
clipping only + input norm	77.1	83.2	82.0	86.6	87.6	87.6	79.3	73.8	82.9	82.4	82.5	85.2
CBMP-Base (no input norm)	75.8	82.0	84.5	81.7	82.2	82.8	74.3	79.0	81.5	76.4	77.8	78.9
per-message RMSNorm	80.7	85.4	86.4	83.6	85.2	85.5	79.7	82.5	83.7	77.6	78.8	81.1
unbounded message	72.9	76.4	79.4	75.8	79.2	80.0	73.5	71.4	77.1	73.6	74.3	75.2
GCN	77.6	78.9	78.1	79.5	80.8	79.4	75.7	76.4	75.5	78.0	77.8	76.5
GCN + input RMSNorm (message stream)	66.4	63.6	65.8	67.2	62.8	64.0	67.8	66.5	66.0	71.0	69.1	69.4
GCN + input RMSNorm (both streams)	90.4	90.6	90.1	88.2	88.4	88.9	87.0	86.8	86.8	88.2	88.0	86.2
GCN, GRB LayerNorm scheme	51.4	83.1	89.6	51.7	81.2	87.8	57.0	81.6	88.8	63.8	81.9	91.8
GCN + radial clip (q=0.95)	77.4	–	76.7	79.1	–	78.4	75.7	–	73.8	78.7	–	76.5
GraphSAGE	79.4	–	79.7	82.1	–	81.0	74.4	–	76.2	74.7	–	73.9
GraphSAGE + input RMSNorm	62.8	–	63.6	67.2	–	60.3	63.4	–	64.4	66.6	–	63.6
single unit, linear	73.7	–	68.4	76.8	–	74.0	73.5	–	67.7	77.0	–	75.2
single unit, tanh	74.6	–	73.1	74.9	–	74.2	69.8	–	68.2	71.8	–	73.3

Table 24: Cost per model (E7; idle GPU; 50 repeats): parameters, forward ms, calibration-pass ms, training-step ms, peak training memory (MB).

model	params	Amazon-ratings				Roman-empire				Tolokers				Computers				ogbn-arxiv			
		fwd	calib.	step	train MB	fwd	calib.	step	train MB	fwd	calib.	step	train MB	fwd	calib.	step	train MB	fwd	calib.	step	train MB
CBMP	74k	1.8	2.1	5.0	279	1.5	1.7	3.9	240	2.7	2.9	6.0	626	2.0	2.1	4.6	392	17.2	18.4	40.1	2351
CBMP, family hp	74k	1.8	2.0	4.5	279	1.5	1.7	3.9	240	2.7	2.9	6.0	626	2.0	2.1	4.7	393	17.2	18.4	40.0	2350
CBMP + budget	74k	1.8	2.3	4.8	285	1.5	2.0	4.2	248	2.8	3.0	6.2	629	2.0	2.3	4.8	396	17.5	20.7	41.6	2423
CBMP-Base (no input norm)	74k	1.6	1.9	4.0	272	1.4	1.6	3.4	229	2.7	2.8	5.8	623	1.9	2.1	4.4	389	16.6	17.9	37.1	2266
unbounded message + input norm	74k	1.4	1.6	3.7	231	1.1	1.4	3.2	174	2.6	2.7	5.6	602	1.8	1.9	4.2	365	14.8	16.1	34.9	1909
unbounded message	74k	1.3	1.5	3.2	224	1.1	1.3	2.7	162	2.5	2.7	5.4	599	1.7	1.9	3.9	361	14.2	15.5	31.9	1867
per-message RMSNorm	74k	1.8	1.8	5.4	302	1.6	1.6	4.8	278	2.8	2.8	6.5	623	2.0	2.1	5.3	389	17.8	17.8	46.5	2433
MLP (no graph)	28k	0.5	0.5	1.5	88	0.4	0.5	1.4	83	0.2	0.2	1.0	52	0.3	0.3	1.3	87	3.2	3.2	9.6	440
GCN	28k	1.0	1.0	2.8	177	0.8	0.8	2.0	110	2.6	2.6	5.9	576	1.6	1.6	3.5	334	11.9	12.0	26.3	1535
GCN + input RMSNorm	28k	1.1	1.1	3.0	182	0.8	0.9	2.7	115	2.6	2.6	5.4	578	1.7	1.7	3.8	337	12.6	12.6	29.3	1577
GCN + input RMSNorm, both streams	28k	1.1	1.1	3.0	182	0.9	0.9	2.5	115	2.6	2.6	5.4	578	1.7	1.7	3.8	337	12.5	12.6	29.3	1576
GCN, GRB LayerNorm scheme	29k	1.2	1.3	3.1	211	1.0	1.0	2.6	141	2.8	2.8	5.6	579	2.0	2.0	4.3	378	14.3	14.3	31.8	1660
GATv2	33k	2.3	2.3	5.5	390	1.6	1.6	3.8	197	7.6	7.7	16.8	1657	4.5	4.5	10.1	855	32.4	32.4	71.3	4695
Soft Median	28k	21.4	21.4	24.9	390	12.1	12.1	14.9	292	104.0	103.9	117.1	1562	54.1	54.1	65.2	941	391.7	391.6	457.6	6014
Soft Median + input RMSNorm	28k	21.5	21.6	25.4	396	12.2	12.2	14.9	304	104.1	104.1	117.4	1564	54.3	54.3	65.5	948	392.3	392.3	460.2	6098
AirGNN (adapted)	28k	3.0	3.0	8.3	369	2.2	2.2	6.0	236	8.6	8.7	27.1	1362	5.2	5.2	22.3	726	41.3	41.3	151.0	5179

Table 25: Full main comparison (test %; 10 seeds, 5 on ogbn-arxiv): clean, mean corruption AUC (noise / scaling / injection, equal weights), feature PGD $\varepsilon = 0.2$, random-connection injection v3 at $\times 1$ and $\times 4$, targeted injection v3 at $\times 1$ and $\times 4$, fixed-norm direction attack.

model	Amazon-r.								Roman-r.								Tolokers								Computers								ogbn-arxiv							
	clean	corr.	PGD	inj. ×1	inj. ×4	targ. ×1	targ. ×4	dir.	clean	corr.	PGD	inj. ×1	inj. ×4	targ. ×1	targ. ×4	dir.	clean	corr.	PGD	inj. ×1	inj. ×4	targ. ×1	targ. ×4	dir.	clean	corr.	PGD	inj. ×1	inj. ×4	targ. ×1	targ. ×4	dir.	clean	corr.	PGD	inj. ×1	inj. ×4	targ. ×1	targ. ×4	dir.
CBMP	47.6	47.3	32.9	37.6	32.3	28.1	27.8	29.9	82.0	80.8	70.4	71.8	71.7	18.3	17.9	66.5	83.2	82.0	80.1	74.1	73.1	68.3	63.0	74.2	90.2	90.0	86.9	87.3	87.3	83.8	83.8	86.2	70.2	68.0	64.4	63.0	61.6	54.5	54.5	59.8
CBMP, family hyper-parameters	47.6	47.3	32.9	37.5	32.5	28.1	27.8	29.9	82.0	80.8	70.4	71.8	71.7	18.3	17.9	66.5	82.5	82.1	79.9	73.7	73.3	67.0	62.6	73.1	89.4	89.2	86.6	86.8	86.8	83.5	83.5	85.8	70.1	68.3	63.1	61.1	61.0	49.9	49.9	58.3
CBMP-Base (calibrated bounded message MLP, no input norm)	48.6	46.7	32.7	32.5	32.5	20.5	20.3	25.1	81.2	79.4	72.9	65.9	65.2	42.6	40.1	63.0	82.6	82.1	79.5	72.7	68.7	64.3	57.5	70.8	90.8	90.3	81.4	81.6	79.9	66.7	62.8	68.4	70.3	69.7	63.4	57.7	56.7	40.0	37.1	49.2
unbounded message MLP + input norm (own tuning)	46.5	45.9	30.4	34.6	34.6	24.4	24.2	27.6	80.9	79.8	68.1	69.3	69.2	52.6	52.1	64.1	82.7	82.0	78.7	73.1	71.5	65.4	61.7	72.3	90.6	90.6	87.1	87.0	87.0	82.6	82.4	86.0	69.9	69.6	64.2	62.6	62.4	56.3	54.3	58.9
unbounded message MLP + input norm (same parameters, family hp)	46.5	45.9	30.4	34.6	34.6	24.4	24.2	27.6	80.9	79.8	68.1	69.3	69.2	52.6	52.1	64.1	82.0	81.5	78.4	72.3	71.3	65.8	61.3	70.0	90.2	90.0	86.2	86.4	86.1	81.4	80.9	84.5	69.0	68.8	62.9	61.7	61.0	50.5	50.5	57.6
unbounded message MLP (same parameters)	47.1	46.3	30.3	34.6	34.6	24.3	24.2	–	81.0	79.8	68.2	69.3	69.2	–	–	64.1	82.7	82.0	78.7	73.1	71.5	65.4	61.7	–	90.6	90.6	87.1	87.0	87.0	82.6	82.4	–	69.9	69.6	64.2	62.6	62.4	–	54.3	58.9	–
GCN	47.1	41.3	30.3	15.4	11.5	13.3	8.3	15.1	80.1	68.9	67.3	41.0	38.8	4.0	3.5	57.8	82.1	79.2	79.3	70.6	57.8	62.1	48.3	64.9	90.3	87.1	86.5	62.1	57.7	17.6	10.5	54.6	69.4	65.9	63.4	43.5	35.0	17.2	8.5	30.8
GCN + input RMSNorm (message stream)	47.3	46.3	31.0	33.7	33.5	21.4	19.8	27.3	79.6	78.5	66.1	67.1	67.0	47.9	47.2	62.0	82.5	82.1	79.4	73.6	72.5	66.0	63.7	73.5	91.1	91.0	85.1	85.3	85.0	76.3	75.3	82.7	69.8	69.6	63.5	62.3	62.2	54.0	53.2	59.0
GCN + input RMSNorm (both streams)	46.6	46.1	30.5	34.8	34.7	24.8	24.5	–	80.4	79.4	67.7	69.4	69.4	53.5	53.2	–	82.9	82.6	80.3	74.6	74.1	68.0	66.2	–	90.3	90.2	87.7	87.7	87.5	84.5	84.2	–	69.7	69.5	64.0	62.6	62.6	54.2	54.2	–
GCN, GRB LayerNorm[†]	47.8	47.2	30.4	35.0	35.0	23.3	22.7	27.1	81.2	80.1	68.4	70.0	69.8	53.0	52.6	64.4	83.4	83.0	80.1	74.2	74.7	68.5	66.4	74.1	90.4	90.3	87.6	87.5	87.4	83.8	83.8	86.7	69.9	69.6	63.5	62.2	62.2	53.9	53.4	58.0
per-message RMSNorm message MLP	49.0	48.3	32.6	36.6	36.2	23.9	23.0	29.8	81.3	80.1	69.3	69.9	69.6	53.2	52.5	64.8	83.0	82.6	80.1	75.3	71.9	64.8	61.9	74.6	91.1	90.6	85.1	83.3	83.0	71.9	70.9	78.1	70.2	70.2	63.1	60.8	60.8	48.1	47.8	55.3
final-message clipping only + input norm	48.1	47.5	30.6	35.3	35.2	24.6	24.0	28.0	80.4	79.2	68.3	69.1	69.0	53.1	53.0	64.2	81.8	81.4	79.3	72.6	70.4	66.7	64.5	70.2	89.6	89.4	86.5	86.8	86.5	82.2	82.7	85.3	69.1	68.8	63.0	61.1	61.1	50.4	50.4	58.3
Soft Median[‡]	46.2	44.3	30.7	30.3	30.3	18.6	18.6	22.4	79.0	77.2	67.0	65.2	65.2	39.4	39.4	60.9	82.0	81.6	78.9	73.6	73.6	65.7	58.5	70.2	89.9	87.9	86.5	75.3	75.2	47.2	46.7	65.5	68.9	68.6	62.1	58.6	58.6	44.9	44.9	53.5
Soft Median + input RMSNorm[‡]	45.3	44.9	31.3	34.7	37.5	20.5	22.6	30.2	78.3	77.4	65.8	68.6	68.5	52.8	52.3	62.4	82.0	81.9	78.5	74.7	74.3	64.9	63.3	72.2	90.6	90.1	87.9	88.0	87.9	83.0	82.9	87.8	68.1	68.0	62.5	62.5	62.5	55.3	55.3	60.2
GraphSAGE	48.5	41.5	–	14.1	9.3	13.3	7.3	–	79.9	68.6	–	39.6	37.1	4.6	3.9	–	82.3	78.7	–	73.4	63.3	67.3	32.8	–	90.6	87.3	–	67.4	59.2	26.8	14.0	–	70.0	66.6	–	50.5	38.5	30.1	12.7	–
GATv2	49.2	41.2	25.4	17.1	16.9	15.3	12.9	13.3	54.9	51.1	51.8	54.3	54.3	13.5	13.5	62.6	83.9	74.3	80.6	63.8	47.1	55.1	42.6	48.6	94.1	95.2	84.7	51.3	36.1	7.3	4.2	39.8	70.7	67.1	62.2	14.8	8.9	10.0	6.0	4.4
AirGNN[§]	42.5	41.7	34.6	23.6	10.7	12.7	6.5	25.6	78.8	70.2	68.6	41.1	19.9	6.1	4.1	59.4	80.1	78.7	78.4	72.8	61.4	65.4	49.0	66.7	87.2	87.1	85.1	82.5	60.2	70.0	24.0	84.4	70.3	67.2	65.3	60.0	6.5	22.2	1.8	35.0
MLP (no graph)	41.9	41.9	51.7	41.9	41.9	41.9	41.9	41.9	65.5	65.5	65.5	65.5	65.5	65.5	65.5	65.5	65.5	65.5	73.1	73.1	73.1	73.1	73.1	73.1	84.0	84.0	84.0	84.0	84.0	84.0	84.0	84.0	53.7	53.7	53.7	53.7	53.7	53.7	53.7	53.7

Table 26: Mechanism measurements under $\times 16$ scaling of 10% of the non-test nodes (three seeds): layer-1 aggregate drift over the Proposition 1 bound (maximum over test nodes; bounded operators only), 99th percentile of the final-message norm, $\|R\|_2$, budget trigger rate (%), relative logit drift (%), prediction-flip rate (%), fraction of test nodes losing more than half their margin (%).

model	Amazon-r.						Roman-r.						Tolokers						Computers						ogbn-arxiv					
	drift/bound	$\|m\|_{\max}$	$\|R\|_2$	trig.	logit	flips	margin	drift/bound	$\|m\|_{\max}$	$\|R\|_2$	trig.	logit	flips	margin	drift/bound	$\|m\|_{\max}$	$\|R\|_2$	trig.	logit	flips	margin	drift/bound	$\|m\|_{\max}$	$\|R\|_2$	trig.	logit	flips	margin	drift/bound	$\|m\|_{\max}$	$\|R\|_2$	trig.	logit	flips	margin
CBMP (ours)	0.23	12.65	3.38	–	6.1	5.0	5.4	0.04	35.58	6.52	–	0.9	0.4	0.4	0.12	36.63	4.86	–	4.2	1.2	2.0	0.25	30.83	3.38	–	4.9	0.7	1.0	0.04	19.32	3.93	–	1.1	0.6	0.7
CBMP without calibration	0.26	14.35	2.95	–	6.1	4.9	5.6	0.13	64.21	6.54	–	1.0	0.6	0.8	0.08	30.38	3.41	–	4.9	3.5	5.4	0.17	10.44	1.34	–	2.2	0.4	0.7	0.10	18.64	2.96	–	2.4	1.3	1.6
CBMP + final-message budget (q=0.95)	0.22	12.57	3.43	5.0	6.0	4.6	5.3	0.04	36.78	6.52	4.8	0.9	0.4	0.5	0.08	27.60	2.98	4.6	5.5	3.3	5.7	0.22	10.67	1.23	5.0	3.2	0.7	1.0	0.10	17.08	2.31	5.0	2.6	1.3	1.7
unbounded message MLP + input norm (same parameters)	–	22.87	1.59	–	7.4	4.8	5.1	–	56.64	3.45	–	0.9	0.5	0.6	–	33.73	2.11	–	6.5	3.0	5.9	–	15.31	1.12	–	4.0	0.8	1.4	–	23.12	1.53	–	2.9	1.5	1.8
final-message clipping only + input norm	–	26.24	2.09	5.0	6.4	4.0	4.5	–	63.96	4.59	4.5	0.9	0.4	0.7	–	41.60	2.95	5.0	5.7	3.0	5.7	–	12.83	1.24	5.0	2.9	0.6	1.0	–	25.67	1.79	5.0	2.6	1.4	1.7
CBMP-Base (calibrated bounded message MLP, no input norm)	0.10	2.88	1.55	–	22.4	15.9	12.9	0.18	32.68	6.63	–	4.9	2.9	3.7	0.14	19.40	2.31	–	5.3	1.6	3.1	0.13	6.48	0.91	–	10.1	1.6	2.6	0.10	3.75	1.22	–	9.3	3.7	5.2
uncalibrated tanh message MLP	0.20	1.81	2.26	–	48.0	30.6	17.4	0.15	17.46	5.33	–	7.1	4.1	5.5	0.33	14.72	3.24	–	17.4	11.8	22.7	0.06	0.19	1.97	–	37.9	9.5	21.7	0.27	3.60	1.81	–	16.6	9.4	10.2
CBMP-Base + budget (no input norm)	0.08	2.88	1.58	5.0	18.7	13.2	12.1	0.15	30.42	6.21	4.5	5.7	3.4	4.3	0.13	19.38	2.27	4.2	5.2	1.8	3.2	0.13	6.59	0.91	5.0	9.3	1.5	2.4	0.07	3.60	1.20	5.0	8.3	3.4	4.7
GCN	–	–	–	–	49.9	30.9	17.1	–	–	–	–	22.5	18.4	17.5	–	–	–	–	37.1	9.9	18.7	–	–	–	–	42.3	10.4	29.1	–	–	–	–	19.5	11.8	13.3
GCN + input RMSNorm (message stream)	–	–	–	–	13.5	12.6	11.5	–	–	–	–	3.7	2.2	2.6	–	–	–	–	7.4	2.5	5.5	–	–	–	–	5.1	0.8	1.2	–	–	–	–	7.6	3.2	4.9
GCN + input RMSNorm (both streams)	–	–	–	–	3.4	2.7	3.3	–	–	–	–	1.0	0.4	0.5	–	–	–	–	3.2	0.8	1.5	–	–	–	–	3.9	0.7	0.9	–	–	–	–	1.4	0.9	1.2
GCN, GRB LayerNorm scheme (reproduction)	–	–	–	–	0.0	0.0	0.0	–	–	–	–	0.0	0.0	0.0	–	–	–	–	0.0	0.0	0.0	–	–	–	–	0.2	0.0	0.1	–	–	–	–	0.0	0.0	0.0
GATv2	–	–	–	–	58.7	36.0	20.7	–	–	–	–	6.6	4.3	4.8	–	–	–	–	63.3	29.2	28.5	–	–	–	–	37.9	9.0	19.6	–	–	–	–	33.8	16.4	26.7

Table 27: Full ablation (test %; clean, random-connection injection ×1, targeted injection ×1; 10 seeds, 5 on ogbn-arxiv): every variant of Section 6.2, the hidden-vector clip, the uncalibrated message MLP and the E11 candidate.

model	Amazon-r.			Roman-e.			Tolokers			Computers			ogbn-arxiv		
	clean	inj.	targ.	clean	inj.	targ.	clean	inj.	targ.	clean	inj.	targ.	clean	inj.	targ.
CBMP	47.6	37.6	28.1	82.0	71.8	58.3	83.2	74.1	66.3	90.2	87.3	83.8	70.2	63.0	54.5
CBMP, family hyper-parameters	47.6	37.6	28.1	82.0	71.8	58.3	82.5	73.3	67.0	89.4	86.9	83.7	69.1	61.1	50.0
CBMP without calibration	47.1	37.2	27.7	83.0	72.4	59.3	82.5	73.2	66.6	89.1	86.8	83.7	69.1	61.1	49.8
CBMP + final-message budget (q=0.95)	47.8	37.8	27.9	82.0	71.7	58.3	82.5	73.4	67.3	89.5	87.0	83.9	69.0	61.1	49.9
CBMP without calibration + budget	47.1	37.3	27.9	83.0	72.7	59.8	82.6	73.5	67.1	89.2	86.9	84.0	69.1	61.1	49.8
unbounded message MLP + input norm (own tuning)	46.3	34.6	24.4	80.9	69.3	52.6	82.7	73.1	65.4	90.7	87.2	82.6	69.9	62.6	54.3
unbounded message MLP + input norm (same parameters, family hp)	46.3	34.6	24.4	80.9	69.3	52.6	82.0	72.5	65.8	90.2	86.4	81.4	69.0	61.1	50.5
final-message clipping only + input norm	46.1	35.4	25.8	80.4	69.1	53.5	81.8	72.4	66.5	89.6	86.8	83.2	69.1	61.1	50.3
CBMP, input norm on the message stream only	48.6	36.6	24.8	80.9	69.7	53.3	82.7	73.1	64.7	91.1	82.5	69.4	70.3	61.1	49.3
message stream only + budget	48.6	36.7	24.3	80.9	69.4	53.1	82.7	73.1	65.1	91.1	83.5	72.9	70.3	61.2	49.4
CBMP + budget after the sum	47.9	37.8	28.0	82.0	71.8	58.0	82.6	73.6	67.4	89.5	86.9	83.6	69.0	61.0	49.6
CBMP-Base (calibrated bounded message MLP, no input norm)	48.6	32.5	20.5	81.2	65.9	42.6	82.6	72.7	64.3	90.8	81.6	66.7	70.3	57.7	40.0
uncalibrated tanh message MLP	45.1	17.4	13.1	82.6	66.9	43.5	81.7	71.8	63.6	89.9	59.5	13.4	68.9	46.7	18.8
CBMP-Base + budget (no input norm)	48.7	35.4	22.4	81.1	66.1	43.1	82.7	72.8	64.6	90.9	83.0	71.3	70.4	59.9	44.8
uncalibrated tanh + budget (no input norm)	45.3	30.3	20.1	82.6	67.1	44.3	81.7	72.3	65.6	89.7	83.3	75.0	69.0	57.9	42.3
final-message clipping only	45.0	30.1	20.1	79.5	63.2	37.0	81.4	71.9	65.1	89.9	83.4	74.5	68.8	57.9	42.7
hidden-vector clipping (before R)	45.4	15.4	19.4	80.1	62.4	34.8	81.4	70.8	61.8	89.8	60.3	21.4	68.9	43.7	15.0
per-message RMSNorm + input norm both streams (E11)	48.3	38.5	28.8	82.2	71.9	58.6	82.7	73.6	67.1	89.5	87.2	84.1	69.4	61.0	49.3
GCN, GRB LayerNorm†	47.8	35.0	23.3	81.2	70.0	53.0	83.4	74.7	66.5	90.4	87.5	83.8	69.9	62.2	53.9

Table 28: Stream × depth × inter-layer LayerNorm ablation on Computers and Roman-empire (test %; clean and targeted injection v3 ×1; 10 seeds; family hyper-parameters). "GCN" rows apply the per-node input RMSNorm to the named stream(s) of GCN; "CBMP + budget" rows apply it to the calibrated bounded message MLP with the final-message budget ($q = 0.95$; the ablation specification was fixed before the budget was dropped from the method, and the budget changes these numbers by at most 0.3, Table 2). "no input norm, 2 layers" of the bounded arm is spec-identical to "CBMP-Base + budget" of Table 27.

base	variant	Computers		Roman-empire	
		clean	targ.	clean	targ.
GCN	message stream L2	91.1	76.2	79.6	47.9
GCN	message stream L2 no inter-layer LN	91.1	78.7	79.4	46.8
GCN	message stream L4	90.9	78.2	82.9	48.3
GCN	message stream L4 no inter-layer LN	90.9	77.9	83.4	51.2
GCN	message stream L2 first layer only	91.1	75.7	79.8	47.7
GCN	residual stream L2	89.3	85.5	79.8	5.9
GCN	residual stream L2 no inter-layer LN	89.7	84.1	79.3	6.0
GCN	residual stream L4	90.5	85.7	83.9	12.1
GCN	residual stream L4 no inter-layer LN	90.9	85.7	85.7	32.4
GCN	residual stream L2 first layer only	89.4	85.1	80.3	6.1
GCN	both streams L2	90.3	84.4	80.4	53.5
GCN	both streams L2 no inter-layer LN	90.4	84.6	81.1	55.6
GCN	both streams L4	90.8	84.6	83.9	55.8
GCN	both streams L4 no inter-layer LN	90.8	84.7	85.5	60.0
GCN	both streams L2 first layer only	90.3	84.1	81.0	53.2
GCN	no input norm L2	90.3	19.7	80.1	4.1
GCN	no input norm L2 no inter-layer LN	87.0	6.6	80.0	3.8
GCN	no input norm L4	90.8	37.6	84.3	6.8
GCN	no input norm L4 no inter-layer LN	89.5	5.8	84.4	5.3
CBMP + budget	message stream L2	91.0	72.7	81.0	53.6
CBMP + budget	message stream L2 no inter-layer LN	91.1	76.5	82.0	53.8
CBMP + budget	message stream L4	91.2	76.0	81.7	55.7
CBMP + budget	message stream L4 no inter-layer LN	91.3	76.8	84.9	59.0
CBMP + budget	message stream L2 first layer only	91.0	72.9	81.0	53.5
CBMP + budget	residual stream L2	89.4	83.1	81.9	49.2
CBMP + budget	residual stream L2 no inter-layer LN	89.3	83.0	82.7	53.0
CBMP + budget	residual stream L4	90.8	83.9	82.0	57.2
CBMP + budget	residual stream L4 no inter-layer LN	90.9	84.0	86.2	59.1
CBMP + budget	residual stream L2 first layer only	89.5	83.4	82.0	47.6
CBMP + budget	both streams L2	89.5	84.0	82.0	57.9
CBMP + budget	both streams L2 no inter-layer LN	89.5	83.9	83.2	58.9
CBMP + budget	both streams L4	90.9	84.3	76.5	55.7
CBMP + budget	both streams L4 no inter-layer LN	91.0	84.3	86.4	62.5
CBMP + budget	both streams L2 first layer only	89.6	84.1	82.2	58.1
CBMP + budget	no input norm L2	90.9	71.3	81.1	43.1
CBMP + budget	no input norm L2 no inter-layer LN	91.0	72.3	82.4	40.9
CBMP + budget	no input norm L4	91.1	74.8	82.5	49.5
CBMP + budget	no input norm L4 no inter-layer LN	91.3	78.5	84.9	47.6