Cost-Sensitive Neighborhood Aggregation for Heterophilous Graphs: When Does Per-Edge Routing Help?

Eyal Weiss[0000-0003-0508-6213]

Computer Science Faculty, Technion — Israel Institute of Technology
eweiss@campus.technion.ac.il

Abstract. Recent work distinguishes two heterophily regimes: *adversarial*, where cross-class edges dilute class signal and harm classification, and *informative*, where the heterophilous structure itself carries useful signal. We ask: *when does per-edge message routing help, and when is a uniform spectral channel sufficient?* To operationalize this question we introduce *Cost-Sensitive Neighborhood Aggregation* (CSNA), a GNN layer that computes pairwise distance in a learned projection and uses it to soft-route each message through concordant and discordant channels with independent transformations. Under a contextual stochastic block model we show that cost-sensitive weighting preserves class-discriminative signal where mean aggregation provably attenuates it, provided $w_+/w_- > q/p$. On six benchmarks with uniform tuning, CSNA is competitive with state-of-the-art methods on adversarial-heterophily datasets (Texas, Wisconsin, Cornell, Actor) but underperforms on informative-heterophily datasets (Chameleon, Squirrel)—precisely the regime where per-edge routing has no useful decomposition to exploit. The pattern is itself the finding: the cost function's ability to separate edge types serves as a diagnostic for the heterophily regime, revealing *when* fine-grained routing adds value over uniform channels and when it does not. Code is available at `https://github.com/eyal-weiss/CSNA-public`.

Keywords: Graph Neural Networks · Heterophily · Message Passing · Cost-Sensitive Aggregation

1 Introduction

Message-passing Graph Neural Networks (GNNs) [6,14,5] aggregate features from a node's neighborhood under an implicit *homophily* assumption: connected nodes share labels or properties. When this assumption holds, aggregation smooths representations within a class and yields strong node classification [15]. However, many real-world graphs exhibit *heterophily*—connected nodes frequently differ in label [10,16]—and on such graphs standard GNNs can perform worse than a simple Multi-Layer Perceptron (MLP) that ignores graph structure entirely, because aggregation *dilutes* rather than *reinforces* class signal [16].

Several architectures address heterophily: H2GCN [16] separates ego from multi-hop neighbor representations; GPRGNN [3] learns polynomial graph filter coefficients; ACM-GNN [8] mixes low-pass, high-pass, and identity channels with per-node gating. These methods share a common intuition—that different edges carry different quality of information for classification—but differ in *how* they distinguish edges. ACM-GNN, the most closely related prior work, uses three fixed spectral channels (aggregation, diversification, identity) that are uniform within each channel: every edge receives the same treatment within a given channel.

We propose *Cost-Sensitive Neighborhood Aggregation* (CSNA), which takes a different approach: compute pairwise distance in a learned projection space, and use this distance to soft-route each message through two channels—concordant (low cost, likely same-class) and discordant (high cost, likely different-class)—each with its own learned transformation. A per-node gating mechanism then combines the channels with an ego (the node's own) representation. The key difference from ACM-GNN is that CSNA's routing is *per-edge*: each edge receives an individualized routing weight based on the learned distance between its endpoints, rather than a uniform spectral filter applied identically to all edges. This finer granularity comes at the cost of additional per-edge computation (3–10× overhead vs. GCN; see Section D).

Contributions. (1) We characterize when per-edge routing helps: CSNA's learned cost function achieves strong edge-type separation on adversarial-heterophily datasets but not on informative-heterophily datasets, operationalizing the regime distinction through a concrete diagnostic (Section 5). (2) We introduce CSNA, a dual-channel message-passing layer with per-edge cost-based routing (Section 3); the default ("lite") version uses only observed divergence g_{ij} in a learned projection; an extended version adds a learned component h_{ij}. (3) Under a contextual stochastic block model (CSBM), we prove that cost-sensitive weighting preserves class signal where mean aggregation degrades it, provided $w_+/w_- > q/p$ (Section 4).

2 Related Work

Heterophily-aware GNNs. Standard GNNs (GCN [6], GAT [14], GraphSAGE [5]) degrade as homophily decreases [16]. H2GCN [16] separates ego from higher-order neighbor embeddings. GPRGNN [3] learns polynomial graph filter coefficients that can model both low-pass and high-pass responses. FAGCN [2] assigns positive or negative attention weights. ACM-GNN [8] is the most closely related method: it decomposes aggregation into three spectral channels (low-pass, high-pass, identity) and combines them with per-node gating. CSNA shares the dual-channel-plus-gating architecture but replaces uniform spectral filters with per-edge learned routing.

Distinction from ACM-GNN. Both ACM-GNN and CSNA route messages through multiple channels and gate the output per-node. The difference is in channel con-

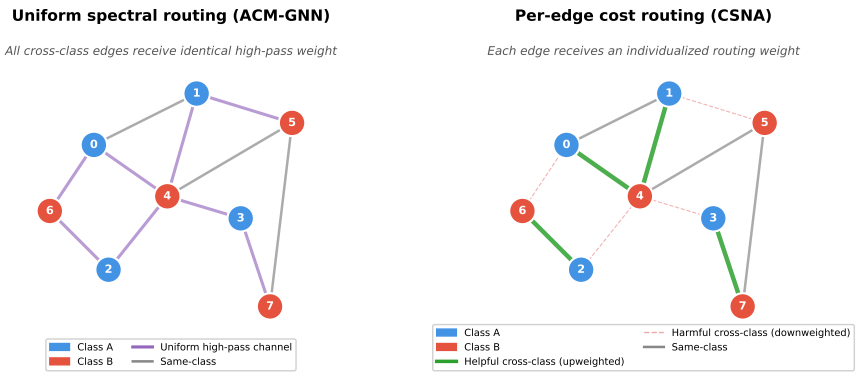

Fig. 1. Toy example: a heterophilous graph where cross-class edges have mixed utility. **Left:** ACM-GNN's uniform high-pass channel assigns identical weight to all cross-class edges. **Right:** CSNA's per-edge cost routing upweights helpful cross-class edges (thick green) and downweights harmful ones (thin dashed red). This distinction is possible only when the cost function can separate edge types—the adversarial-heterophily regime.

struction: ACM-GNN's channels are defined by fixed spectral operations (mean aggregation for low-pass, signed aggregation for high-pass), so every edge within a channel is treated identically. CSNA computes a per-edge concordance score from learned pairwise distance and uses it to route each edge independently. This is finer-grained but more expensive: ACM-GNN has the same asymptotic cost as GCN ($O(|E| \cdot d)$ per channel, but with fixed structure), while CSNA requires an additional $O(|E| \cdot d)$ distance computation.

When per-edge routing helps: a toy example. Figure 1 illustrates the key scenario where per-edge routing outperforms uniform spectral channels. Consider a heterophilous graph in which most edges are cross-class, but the cross-class edges have *mixed utility*: some neighbors' features, when transformed, carry complementary information that aids classification, while others are misleading. A uniform high-pass channel (as in ACM-GNN) treats all cross-class edges identically—it cannot distinguish helpful from harmful heterophilous neighbors. CSNA's per-edge cost function assigns an individual routing weight to each edge, so the model can upweight informative cross-class edges and downweight misleading ones. This advantage materializes when node features within a class are not homogeneous—some cross-class neighbors provide complementary signal while others contribute noise—precisely the adversarial-heterophily regime. On informative-heterophily datasets, where nearly all heterophilous edges carry useful structural signal, the per-edge decomposition has no useful separation to exploit and the extra routing overhead buys nothing.

Over-smoothing and over-squashing. These pathologies [7,9,1,13] are compounded under heterophily. Cost-based routing reduces cross-class smoothing but does not address over-squashing, which is a topological bottleneck issue.

3 Method

3.1 Preliminaries

Let $\mathcal{G} = (V, E)$ be an undirected graph with $n = |V|$ nodes and feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. Each node i has a label $y_i \in \{1, \ldots, C\}$. The *edge homophily ratio* is $\mathcal{H} = |\{(i,j) \in E : y_i = y_j\}|/|E|$.

In standard message-passing [4], node i's representation at layer $\ell + 1$ is:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}\, \mathbf{W}^{(\ell)}\, \mathbf{h}_j^{(\ell)} \Big), \tag{1}$$

where $\mathcal{N}(i)$ includes i itself, α_{ij} are aggregation weights, and σ is a nonlinearity.

3.2 Cost-Sensitive Neighborhood Aggregation

The core idea is simple: compute pairwise distance in a learned projection, use it to soft-route messages through two channels.

Step 1: Edge cost (observed divergence). For each edge (i, j) at layer ℓ, we compute a distance in a learned projection:

$$g_{ij}^{(\ell)} = \|\mathbf{W}_g\, \mathbf{h}_i^{(\ell)} - \mathbf{W}_g\, \mathbf{h}_j^{(\ell)}\|_2, \tag{2}$$

where $\mathbf{W}_g \in \mathbb{R}^{d' \times d}$ is a learned projection matrix. Low g_{ij} indicates the endpoints are similar in the projected space (likely concordant); high g_{ij} indicates divergence (likely discordant). This is the only cost component in the default (lite) version of CSNA.

Step 2: Concordance routing. The cost is converted to a soft concordance score:

$$s_{ij}^{(\ell)} = \sigma\left(\frac{-g_{ij}^{(\ell)}}{\tau} \right) \in (0, 1), \tag{3}$$

where $\tau > 0$ is a temperature parameter and σ is the sigmoid function. High concordance (low cost) routes the message toward the concordant channel; low concordance (high cost) routes it toward the discordant channel.

Step 3: Dual-channel aggregation. Messages are routed through two channels with independent transformations:

$$\mathbf{h}_i^{\text{con}} = \sum_{j \in \mathcal{N}(i)} \tilde{s}_{ij}\, \mathbf{W}_{\text{con}}\, \mathbf{h}_j^{(\ell)}, \tag{4}$$

$$\mathbf{h}_i^{\text{dis}} = \sum_{j \in \mathcal{N}(i)} \tilde{d}_{ij}\, \mathbf{W}_{\text{dis}}\, \mathbf{h}_j^{(\ell)}, \tag{5}$$

where $\tilde{s}_{ij} = \text{softmax}_{j \in \mathcal{N}(i)}(s_{ij})$, $\tilde{d}_{ij} = \text{softmax}_{j \in \mathcal{N}(i)}(1 - s_{ij})$, and $\mathbf{W}_{\text{con}}, \mathbf{W}_{\text{dis}} \in \mathbb{R}^{d' \times d}$ are independent weight matrices. The concordant channel emphasizes likely same-class neighbors; the discordant channel processes likely different-class neighbors through a separate transformation.

Step 4: Gated combination. The final output combines both channels with the ego representation via per-node gating:

$$\mathbf{h}_i^{(\ell+1)} = \sum_{k \in \{\text{con,dis,self}\}} \gamma_k(\mathbf{h}_i) \cdot \mathbf{h}_i^k, \tag{6}$$

where $\mathbf{h}_i^{\text{self}} = \mathbf{W}_{\text{self}}\, \mathbf{h}_i^{(\ell)}$ and $\gamma(\mathbf{h}_i) = \text{softmax}(\mathbf{W}_\gamma[\mathbf{h}_i^{\text{con}} \| \mathbf{h}_i^{\text{dis}} \| \mathbf{h}_i^{\text{self}}]) \in \mathbb{R}^3$.

Extended variant: $g + h$. An optional extension adds a learned cost component:

$$f_{ij}^{(\ell)} = g_{ij}^{(\ell)} + h_{ij}^{(\ell)}, \quad \text{where} \quad h_{ij}^{(\ell)} = \text{softplus}\big(\mathbf{a}^\top [\mathbf{W}_g \mathbf{h}_i^{(\ell)} \| \mathbf{W}_g \mathbf{h}_j^{(\ell)}]\big), \tag{7}$$

where $[\cdot\|\cdot]$ denotes vector concatenation, $\mathbf{a} \in \mathbb{R}^{2d'}$ is a learnable parameter vector, and f_{ij} replaces g_{ij} in Equation (3). In our experiments, the extended variant wins clearly only on Wisconsin (84.1 vs. 79.6); on all other datasets the lite version performs comparably or better (Section A). We therefore present the lite version (g_{ij} only) as the default.

Calibration regularization. When training with labels, we add a regularizer that penalizes cost overestimation on same-class edges:

$$\mathcal{L}_{\text{cal}} = \frac{1}{|E_{\text{train}}|} \sum_{(i,j) \in E_{\text{train}}} \big[\text{ReLU}(g_{ij} - \mathbb{1}[y_i \neq y_j])\big]^2, \tag{8}$$

where E_{train} denotes edges between labeled nodes. The full objective is $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{cal}}\, \mathcal{L}_{\text{cal}}$, with $\lambda_{\text{cal}} = 0.1$ fixed across all experiments. Note that g_{ij} and the binary indicator $\mathbb{1}[y_i \neq y_j]$ are on different scales; the regularizer acts as a soft penalty that directly shapes the routing signal, encouraging lower costs (and thus higher concordance) for same-class edges.

Algorithm 1 CSNA Layer (Lite Version)

Require: Node features $\mathbf{H}^{(\ell)} \in \mathbb{R}^{n \times d}$, edge index E, temperature τ
Ensure: Updated features $\mathbf{H}^{(\ell+1)} \in \mathbb{R}^{n \times d'}$
for each edge $(i, j) \in E$ **do**
  $g_{ij} \leftarrow \|\mathbf{W}_g \mathbf{h}_i - \mathbf{W}_g \mathbf{h}_j\|_2$ {Pairwise distance in learned projection}
  $s_{ij} \leftarrow \sigma(-g_{ij}/\tau)$ {Concordance score}
end for
for each node $i \in V$ **do**
  $\mathbf{h}_i^{\text{con}} \leftarrow \sum_{j \in \mathcal{N}(i)} \tilde{s}_{ij}\, \mathbf{W}_{\text{con}} \mathbf{h}_j$ {Concordant channel}
  $\mathbf{h}_i^{\text{dis}} \leftarrow \sum_{j \in \mathcal{N}(i)} \tilde{d}_{ij}\, \mathbf{W}_{\text{dis}} \mathbf{h}_j$ {Discordant channel}
  $\mathbf{h}_i^{\text{self}} \leftarrow \mathbf{W}_{\text{self}} \mathbf{h}_i$ {Ego transform}
  $\mathbf{h}_i^{(\ell+1)} \leftarrow \sum_k \gamma_k(\mathbf{h}_i) \cdot \mathbf{h}_i^k$ {Gated combination}
end for

Architecture details. We apply an input MLP before the first CSNA layer, add residual connections, and initialize the gate bias to $[0, 0, 1]$ (favoring the ego channel so the model starts near MLP-like behavior). Self-loops are added before cost computation; they receive $g_{ii} = 0$, so ego information flows primarily through the concordant and self channels. The softmax normalization of concordance weights is performed per source node; we also tested per-destination normalization (more common in message-passing GNNs) and found no consistent difference across datasets.

The CSNA layer is summarized in Algorithm 1.

4 Theoretical Analysis

We analyze the advantage of cost-sensitive aggregation over mean aggregation in heterophilous settings using the *contextual stochastic block model* (CSBM) [16]. Our analysis uses the binary ($C = 2$) case for tractability; our experimental datasets have $C = 5$, and we discuss the multi-class extension in Section F. Importantly, the theorems below apply to *any* weighted aggregation scheme—they are not specific to the $g + h$ decomposition or the lite variant.

Definition 1 (Contextual Stochastic Block Model). *A graph $\mathcal{G}$ is drawn from* $\text{CSBM}(n, 2, p, q, \mu)$ *with n nodes in two equal-sized classes V_+, V_-. Edges are drawn independently: probability p within classes, probability q between classes. Node features: $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{y_i}, \mathbf{I}_d)$, with $\boldsymbol{\mu}_+ = +\frac{\mu}{2}\mathbf{e}_1$ and $\boldsymbol{\mu}_- = -\frac{\mu}{2}\mathbf{e}_1$.*

The homophily ratio is $\mathcal{H} = p/(p + q)$. The heterophilous regime is $q > p$, i.e., $\mathcal{H} < 1/2$.

Theorem 1 (Signal degradation under mean aggregation). *Let $\mathcal{G} \sim \text{CSBM}(n, 2, p, q, \mu)$ with $q > p$. After one round of mean aggregation $\mathbf{H}^{(1)} = \tilde{\mathbf{A}}\mathbf{X}$, where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathbf{A} + \mathbf{I})\mathbf{D}^{-1/2}$ is the symmetrically normalized adjacency, the*

expected between-class separation satisfies:

$$\frac{\mathbb{E}\big[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\big]}{\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2} = \left(\frac{p-q}{p+q}\right)^2 + O\left(\frac{1}{\sqrt{n}}\right), \tag{9}$$

where $\bar{\mathbf{h}}_c^{(1)}$ is the mean representation of class c. When $q > p$, the leading factor is strictly less than 1; when $q \gg p$, it approaches zero.

Proof. For a node $i \in V_+$, same-class neighbors number $n_s(i) \sim \mathrm{Bin}(n/2-1, p)$ and different-class neighbors $n_d(i) \sim \mathrm{Bin}(n/2, q)$. Under mean aggregation:

$$\mathbb{E}[\mathbf{h}_i^{(1)} | y_i = +1] = \frac{p}{p+q}\boldsymbol{\mu}_+ + \frac{q}{p+q}\boldsymbol{\mu}_- + O(1/\sqrt{n}) = \frac{p-q}{p+q}\boldsymbol{\mu}_+ + O(1/\sqrt{n}), \tag{10}$$

where the $O(1/\sqrt{n})$ term follows from concentration of $n_s(i)/d_i$ around $p/(p+q)$ (Hoeffding's inequality, $d_i = \Theta(n)$). By symmetry, $\mathbb{E}[\mathbf{h}_i^{(1)} | y_i = -1] = \frac{p-q}{p+q}\boldsymbol{\mu}_- + O(1/\sqrt{n})$. Averaging over $n/2$ nodes per class and applying the law of large numbers:

$$\mathbb{E}\big[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\big] = \left(\frac{p-q}{p+q}\right)^2 \|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2 + O(1/\sqrt{n}). \square \tag{11}$$

Theorem 2 (Signal preservation under cost-sensitive aggregation). *Under the same CSBM, suppose an aggregation scheme weights edge (i, j) by w_{ij}, with $\mathbb{E}[w_{ij} | y_i = y_j] = w_+$ and $\mathbb{E}[w_{ij} | y_i \neq y_j] = w_-$, where $w_+ > w_- \geq 0$. Then:*

$$\frac{\mathbb{E}\big[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\big]}{\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2} = \left(\frac{p\,w_+ - q\,w_-}{p\,w_+ + q\,w_-}\right)^2 + O(1/\sqrt{n}). \tag{12}$$

This exceeds the mean-aggregation factor $\left(\frac{p-q}{p+q}\right)^2$ if and only if $w_+/w_- > q/p$.

Proof. Replacing uniform weights with w_+ (same-class) and w_- (different-class) in the proof of Theorem 1:

$$\mathbb{E}[\mathbf{h}_i^{(1)} | y_i = +1] \propto p\,w_+\boldsymbol{\mu}_+ + q\,w_-\boldsymbol{\mu}_- + O(1/\sqrt{n}). \tag{13}$$

The between-class scatter follows by substituting $p \to p\,w_+$ and $q \to q\,w_-$:

$$\mathbb{E}\big[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\big] = \left(\frac{p\,w_+ - q\,w_-}{p\,w_+ + q\,w_-}\right)^2 \|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2 + O(1/\sqrt{n}). \tag{14}$$

The condition $w_+/w_- > q/p$ is equivalent to $p\,w_+ > q\,w_-$, ensuring the same-class contribution dominates. $\square$

Remark 1 (Applicability to CSNA). Theorem 2 applies to CSNA's concordant channel with $w_{ij} = s_{ij}$. The condition $s_+/s_- > q/p$ requires the cost function to assign sufficiently higher concordance to same-class edges. The calibration regularizer (Equation (8)) encourages this. However, Theorem 2 is *conditional*: it shows what happens *if* the cost function achieves good separation, not that CSNA *will* learn such separation.

Table 1. Dataset statistics. $\mathcal{H}$ is the edge homophily ratio.

Dataset	Nodes	Edges	Features	Classes	$\mathcal{H}$
Texas	183	287	1,703	5	0.09
Wisconsin	251	458	1,703	5	0.19
Cornell	183	278	1,703	5	0.13
Actor	7,600	26,705	932	5	0.22
Chameleon	2,277	31,396	2,325	5	0.23
Squirrel	5,201	198,423	2,089	5	0.22

Remark 2 (Scope and limitations). (i) The CSBM analysis is for $C = 2$; experiments have $C = 5$. The qualitative conclusion carries over (Section F), but the quantitative bound changes. (ii) Theorem 2 bounds between-class scatter only, not within-class scatter (Section G). (iii) The analysis covers a single layer; multi-layer interactions between evolving representations and the cost function are not analyzed.

5 Experiments

5.1 Datasets

We evaluate on six heterophily benchmarks (Table 1). Texas, Wisconsin, and Cornell are webpage graphs from WebKB [10]. Chameleon and Squirrel are Wikipedia article networks [10]. Actor is a co-occurrence network from film databases [10]. We note that Chameleon and Squirrel have known data quality issues (duplicate nodes) [11].

5.2 Setup

Baselines. We compare against: (1) **MLP** (no graph structure), (2) **GCN** [6], (3) **GAT** [14], (4) **GraphSAGE** [5], (5) **H2GCN** [16], (6) **GPRGNN** [3], and (7) **ACM-GNN** [8].

Protocol. We use 10 random 60%/20%/20% splits (seed 42). *All methods* are tuned over the same hyperparameter grid: learning rate $\in \{0.01, 0.005\}$, hidden dimension $\in \{64, 128\}$. CSNA additionally tunes temperature $\tau \in \{0.1, 0.5, 1.0, 2.0\}$. All models use 2 layers, dropout 0.5, Adam with weight decay 5×10^{-4}, and early stopping (patience 50, max 300 epochs). We save the checkpoint with best validation accuracy and evaluate it *once* on test. We report mean accuracy over 10 splits. We report classification accuracy (fraction of correctly labeled test nodes), averaged over the 10 splits, with standard deviation reflecting split-to-split variability. CSNA here is the lite version (g_{ij} only, no sampling). Complete details in Section H.

Table 2. Node classification accuracy (%) on heterophily benchmarks. All methods tuned over the same grid. Best in **bold**, second-best underlined.

Method	Texas $\mathcal{H}$=0.09	Wisconsin $\mathcal{H}$=0.19	Cornell $\mathcal{H}$=0.13	Actor $\mathcal{H}$=0.22	Chameleon $\mathcal{H}$=0.23	Squirrel $\mathcal{H}$=0.22
MLP	77.3±4.6	83.7±4.8	72.2±3.6	35.0±1.4	51.9±1.8	34.8±1.4
GCN	55.7±9.9	50.6±8.5	47.0±8.7	27.3±1.4	**67.3±1.7**	**53.4±0.8**
GAT	50.8±9.8	51.4±7.8	47.3±5.8	28.0±1.4	65.7±2.2	50.4±1.4
GraphSAGE	76.5±6.8	75.9±5.8	66.2±7.7	34.1±0.6	63.9±2.1	45.8±1.4
H2GCN	75.9±5.0	76.1±6.1	67.8±7.7	31.3±1.7	51.5±2.8	37.3±2.8
GPRGNN	**78.1±8.0**	78.6±5.1	70.0±6.7	34.2±0.7	58.5±2.8	38.4±2.3
ACM-GNN	75.9±6.3	80.0±6.1	69.5±5.1	33.5±1.6	59.3±2.0	41.6±1.3
CSNA (ours)	77.0±8.3	79.6±6.2	**72.7±4.3**	**35.7±1.2**	54.6±2.7	37.8±1.9

5.3 Main Results

Results are in Table 2.

The results reveal a clear split between two types of heterophily benchmarks:

Adversarial heterophily (Texas, Wisconsin, Cornell, Actor). On these datasets, cross-class edges are adversarial: aggregating neighbor features degrades classification. MLP is a strong baseline, and standard GNNs (GCN, GAT) perform poorly. CSNA is competitive with the best methods: it achieves the highest mean accuracy on Cornell and Actor (though the margins over MLP are within one standard deviation), and is within 1–4 points of the best on Texas and Wisconsin. GPRGNN and ACM-GNN are also strong on these datasets.

Informative heterophily (Chameleon, Squirrel). On these datasets, heterophilous structure itself carries useful signal. GCN and GAT substantially outperform all heterophily-specific methods (CSNA, H2GCN, GPRGNN, ACM-GNN). This is consistent with prior observations [11,8]: methods that down-weight cross-class edges lose the informative signal. CSNA's cost-based routing cannot distinguish "harmful" from "useful" heterophilous edges. This distinction—between adversarial and informative heterophily—is increasingly recognized [8,11] and suggests that no single approach dominates all heterophily regimes. CSNA's cost semantics make the distinction explicit: on adversarial datasets, learned costs successfully separate edge types (Fig. 2); on informative datasets, they cannot.

Comparison with ACM-GNN. ACM-GNN and CSNA show a similar overall pattern. CSNA slightly outperforms ACM-GNN on Cornell (+3.2pp) and Actor (+2.2pp), while ACM-GNN is ahead on Chameleon (+4.7pp) and Squirrel (+3.8pp). The accuracy differences between CSNA and ACM-GNN are generally within one standard deviation, suggesting that the per-edge granularity does not consistently translate to measurable accuracy gains over uniform spectral channels. The methods differ mainly in mechanism, not in overall performance level.

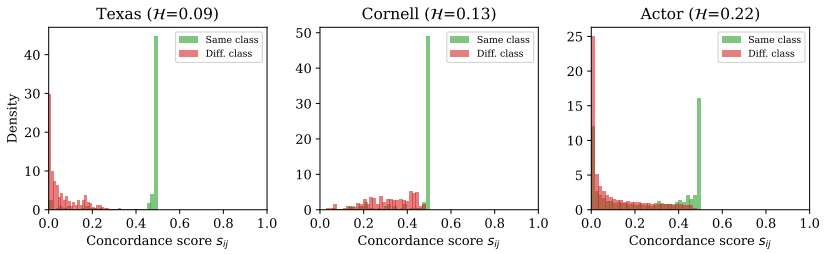

Fig. 2. Learned concordance scores s_{ij} for same-class (green) and different-class (red) edges on three datasets. On adversarial-heterophily datasets (Texas, Cornell), the distributions are well separated; on Actor, overlap is larger. Generated using the extended $(g + h)$ variant; similar separation is observed with the lite version.

Routing quality as a diagnostic. To quantify the cost function's ability to discriminate edge types, we compute the AUC of the concordance score s_{ij} as a binary classifier for same-class vs. different-class edges. On the adversarial-heterophily datasets, AUC ranges from 0.48 (Texas) to 0.83 (Wisconsin), while on informative-heterophily datasets it is 0.55–0.57 (Chameleon, Squirrel). Interestingly, CSNA's accuracy gain over GCN does not correlate simply with AUC: Texas has near-random AUC (0.48) yet CSNA outperforms GCN by 21pp, while Wisconsin has the highest AUC (0.83) and the largest gain (+29pp). This suggests that CSNA's benefit comes not only from edge-type separation but also from the dual-channel architecture and gating mechanism, which can learn useful representations even when the cost function's discrimination is weak. On informative-heterophily datasets, both the low AUC and the poor accuracy confirm that the cost function cannot usefully decompose the neighborhood.

Ablation highlights. We tested four CSNA variants in a factorial design (Section A): the extended model $(g + h)$ vs. the lite default $(g$ only), each with and without stochastic edge sampling [12]. The extended variant wins clearly only on Wisconsin (84.1 vs. 79.6); on all other datasets the lite version matches or exceeds it. Edge sampling slightly reduces accuracy (1–3pp) but provides regularization and scaling benefits. These results motivate presenting the lite version without sampling as the default.

6 Discussion

Limitations—and what they reveal.

– **Homophilous graphs.** CSNA underperforms GCN by 4–11pp on Cora, CiteSeer, and PubMed (Section B). The per-edge routing is not justified when standard aggregation suffices; this is expected, since the cost function has little to separate when most edges are already same-class.

- **Informative heterophily.** CSNA underperforms GCN on Chameleon (by 12.7pp) and Squirrel (by 15.6pp). Rather than a mere negative result, this failure is diagnostic: it identifies datasets where heterophilous edges are uniformly informative and no per-edge decomposition can improve on uniform aggregation. The pattern—success on adversarial heterophily, failure on informative heterophily—is itself a contribution, as it operationalizes the regime distinction through CSNA's cost semantics.
- **Computational cost.** CSNA is 3–10× slower than GCN (Section D), though it uses fewer parameters.
- **Similarity to ACM-GNN.** CSNA shares the dual-channel-plus-gating design with ACM-GNN [8]. The per-edge routing is a meaningful architectural difference, but accuracy gains over ACM-GNN are not consistently significant.

7 Conclusion

We presented CSNA, a GNN layer that computes pairwise distance in a learned projection and uses it to soft-route messages through concordant and discordant channels, with per-node gating. The key architectural difference from the closely related ACM-GNN is per-edge routing rather than uniform spectral channels—a finer-grained mechanism that, in our experiments, yields comparable accuracy at higher computational cost. Our theoretical analysis provides a clean condition ($w_+/w_- > q/p$) under which cost-sensitive weighting improves over mean aggregation. The clearest open question is whether per-edge routing can be adapted to the informative-heterophily regime—where CSNA currently fails—possibly by learning to leverage rather than suppress cross-class signal.

References

1. Alon, U., Yahav, E.: On the bottleneck of graph neural networks and its practical implications. In: International Conference on Learning Representations (ICLR) (2021)
2. Bo, D., Wang, X., Shi, C., Shen, H.: Beyond low-frequency information in graph convolutional networks. In: AAAI Conference on Artificial Intelligence (2021)
3. Chien, E., Peng, J., Li, P., Milenkovic, O.: Adaptive universal generalized pagerank graph neural network. In: International Conference on Learning Representations (ICLR) (2021)
4. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning (ICML) (2017)
5. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
6. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
7. Li, Q., Han, Z., Wu, X.M.: Deeper insights into graph convolutional networks for semi-supervised learning. In: AAAI Conference on Artificial Intelligence (2018)

8. Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.W., Precup, D.: Revisiting heterophily for graph neural networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
9. Oono, K., Suzuki, T.: Graph neural networks exponentially lose expressive power for node classification. In: International Conference on Learning Representations (ICLR) (2020)
10. Pei, H., Wei, B., Chang, B., Lei, Y., Yang, B.: Geom-GCN: Geometric graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2020)
11. Platonov, O., Kuznedelev, D., Diskin, M., Babenko, A., Prokhorenkova, L.: A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In: International Conference on Learning Representations (ICLR) (2023)
12. Rong, Y., Huang, W., Xu, T., Huang, J.: DropEdge: Towards deep graph convolutional networks on node classification. In: International Conference on Learning Representations (ICLR) (2020)
13. Topping, J., Di Giovanni, F., Chamberlain, B.P., Dong, X., Bronstein, M.M.: Understanding over-squashing and bottlenecks on graphs via curvature. In: International Conference on Learning Representations (ICLR) (2022)
14. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (ICLR) (2018)
15. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021)
16. Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., Koutra, D.: Beyond homophily in graph neural networks: Current limitations and effective designs. In: Advances in Neural Information Processing Systems (NeurIPS) (2020)

Appendix

A CSNA Variant Comparison

We compare four CSNA variants in a factorial design: full ($g+h$) vs. lite (g only) $\times$ edge sampling vs. no sampling. Results are in Table 3.

Table 3. CSNA variant comparison: accuracy (%) on all six datasets. "Full" includes both g_{ij} and h_{ij}; "lite" uses g_{ij} only. "Samp" applies edge sampling during training.

Variant	Texas	Wisconsin	Cornell	Actor	Chameleon	Squirrel
Full, no samp	77.0	**84.1**	70.8	35.6	53.0	38.2
Full + samp	74.1	79.8	70.5	**36.3**	51.8	37.4
Lite, no samp	**77.0**	79.6	**72.7**	35.7	**54.6**	**37.8**
Lite + samp	75.7	79.6	70.3	35.8	53.5	37.0

Discussion. The full variant ($g + h$) wins clearly only on Wisconsin (84.1 vs. 79.6), where the learned component h_{ij} provides a useful correction beyond observed divergence. On all other datasets, the lite version matches or exceeds the full version, suggesting that feature divergence g_{ij} alone is a sufficient routing signal. Since h_{ij} introduces additional learnable parameters without a clear consistent advantage, it risks overfitting—particularly on the small datasets in our benchmark suite. We therefore use the lite version (g_{ij} only) as the default.

Edge sampling slightly hurts accuracy on most datasets (1–3pp) but may act as a regularizer on Actor (36.3 vs. 35.6 for the full variant). The accuracy cost of sampling is modest, making it a viable strategy for scaling to larger graphs.

B Homophily Benchmarks

To understand CSNA's behavior across the homophily spectrum, we evaluate on three standard homophilous datasets (Table 4).

Table 4. Accuracy (%) on homophilous benchmarks. CSNA underperforms GCN by 4–11pp.

Method	Cora	CiteSeer	PubMed
GCN	**77.9**	**65.7**	**76.0**
CSNA	67.0	56.3	72.2

CSNA underperforms GCN by 10.9pp on Cora, 9.4pp on CiteSeer, and 3.8pp on PubMed. This is expected: on homophilous graphs, standard aggregation is already effective, and the per-edge routing overhead adds complexity without benefit. The cost function cannot improve on uniform aggregation when most edges are already same-class. These results suggest CSNA's per-edge routing overhead is not justified on homophilous graphs, where standard aggregation already performs well.

C Gate Weight Analysis

Table 5 shows the average gate weights at layer 0, revealing dataset-dependent routing strategies.

The gate weights reveal three distinct strategies:

- **Self-dominant (Wisconsin, Actor):** The model ignores graph structure entirely ($\gamma_{\text{self}} = 1.0$), effectively reducing to MLP. This is consistent with CSNA matching MLP-level accuracy on these datasets. The collapse to MLP-like behavior is itself informative: it indicates that on these datasets, the cost function does not find a useful decomposition of the neighborhood, and the gating mechanism correctly learns to ignore the graph channels.

Table 5. Average gate weights γ across nodes (layer 0).

Dataset	γ_{con}	γ_{dis}	γ_{self}
Texas	0.30	0.13	0.57
Wisconsin	0.00	0.00	1.00
Cornell	0.12	0.28	0.60
Actor	0.00	0.00	1.00
Chameleon	0.34	0.66	0.00
Squirrel	0.14	0.86	0.00

- **Mixed (Texas, Cornell):** The model uses a combination of ego and graph channels. On Cornell, the discordant channel receives substantial weight ($\gamma_{\text{dis}} = 0.28$), indicating that different-class neighbor information is actively processed.
- **Graph-dominant (Chameleon, Squirrel):** The model relies entirely on graph channels, with the discordant channel dominant ($\gamma_{\text{dis}} = 0.66$ and 0.86 respectively). Ironically, these are the datasets where CSNA underperforms GCN, suggesting the discordant channel's separate transformation does not capture the useful heterophilous signal as effectively as GCN's uniform aggregation.

D Runtime and Parameter Comparison

Table 6. Training time (seconds per split) and parameter count. All on 8-core CPU.

Method	Texas	Actor	Chameleon	Squirrel	Params (Texas)
MLP	0.2	3	1	2	219K
GCN	0.2	4	7	62	219K
GAT	0.3	11	20	152	110K
GraphSAGE	0.3	15	30	132	219K
H2GCN	0.6	11	27	327	318K
GPRGNN	0.6	9	14	111	219K
CSNA	1	34	37	205	144K

CSNA is 3–10× slower than GCN due to the per-edge distance computation. On the largest dataset (Squirrel, 198K edges), a single split takes ∼205 seconds vs. 62 for GCN. However, CSNA uses fewer parameters (144K vs. 219K for GCN on Texas) because the projection $\mathbf{W}_g$ is shared between cost computation and the channels. For scaling to larger graphs, edge sampling [12] is a natural option; our ablation (Section A) shows it costs only 1–3pp in accuracy.

E Proof Details

E.1 Detailed Proof of Theorem 1

We provide the complete derivation with explicit concentration bounds. Consider the binary CSBM with n nodes, two equal-sized classes, intra-class edge probability p, and inter-class edge probability $q > p$.

Let $V_+ = \{i : y_i = +1\}$ and $V_- = \{i : y_i = -1\}$, each of size $n/2$. Features: $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{y_i}, \mathbf{I}_d)$, with $\boldsymbol{\mu}_+ = +\frac{\mu}{2}\mathbf{e}_1$ and $\boldsymbol{\mu}_- = -\frac{\mu}{2}\mathbf{e}_1$.

For a node $i \in V_+$: $n_s(i) \sim \text{Bin}(n/2 - 1, p)$ and $n_d(i) \sim \text{Bin}(n/2, q)$, so $d_i = n_s(i) + n_d(i)$.

By Hoeffding's inequality:

$$\Pr\left[\left|\frac{n_s(i)}{d_i} - \frac{p}{p+q}\right| > \epsilon\right] \leq 2\exp\left(-2\epsilon^2 d_i\right), \tag{15}$$

and since $\mathbb{E}[d_i] = (n/2-1)p + (n/2)q = \Theta(n)$, we have $n_s(i)/d_i = p/(p+q) + O_p(1/\sqrt{n})$.

Under mean aggregation:

$$\mathbf{h}_i^{(1)} = \frac{1}{d_i}\sum_{j \in \mathcal{N}(i)} \mathbf{x}_j = \frac{n_s(i)}{d_i}\bar{\mathbf{x}}_+(i) + \frac{n_d(i)}{d_i}\bar{\mathbf{x}}_-(i). \tag{16}$$

Taking expectations:

$$\mathbb{E}[\mathbf{h}_i^{(1)} | y_i = +1] = \frac{p}{p+q}\boldsymbol{\mu}_+ + \frac{q}{p+q}\boldsymbol{\mu}_- + O(1/\sqrt{n}) \tag{17}$$

$$= \frac{p-q}{p+q} \cdot \frac{\mu}{2}\mathbf{e}_1 + O(1/\sqrt{n}). \tag{18}$$

By symmetry, $\mathbb{E}[\mathbf{h}_i^{(1)} | y_i = -1] = \frac{p-q}{p+q} \cdot (-\frac{\mu}{2}\mathbf{e}_1) + O(1/\sqrt{n})$.

The class means $\bar{\mathbf{h}}_c^{(1)}$ concentrate at rate $O(1/\sqrt{n})$:

$$\mathbb{E}\left[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\right] = \left(\frac{p-q}{p+q}\right)^2 \mu^2 + O(1/\sqrt{n}). \tag{19}$$

Since $\|\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\|^2 = \mu^2$, the multiplicative attenuation is $\left(\frac{p-q}{p+q}\right)^2$. □

E.2 Detailed Proof of Theorem 2

The weighted aggregation assigns weight w_{ij} to each edge. For $i \in V_+$:

$$\mathbf{h}_i^{(1)} \propto \sum_{j \in \mathcal{N}(i)} w_{ij}\, \mathbf{x}_j. \tag{20}$$

Taking expectations:

$$\mathbb{E}[\mathbf{h}_i^{(1)}|y_i = +1] \propto \mathbb{E}[n_s(i)] \cdot w_+ \cdot \boldsymbol{\mu}_+ + \mathbb{E}[n_d(i)] \cdot w_- \cdot \boldsymbol{\mu}_- + O(1/\sqrt{n}) \tag{21}$$

$$\propto p\, w_+ \boldsymbol{\mu}_+ + q\, w_- \boldsymbol{\mu}_- + O(1/\sqrt{n}) \tag{22}$$

$$= \frac{p\, w_+ - q\, w_-}{2}\, \mu\, \mathbf{e}_1 + O(1/\sqrt{n}). \tag{23}$$

By the same concentration argument:

$$\mathbb{E}\Big[\|\bar{\mathbf{h}}_+^{(1)} - \bar{\mathbf{h}}_-^{(1)}\|^2\Big] = \left(\frac{p\, w_+ - q\, w_-}{p\, w_+ + q\, w_-}\right)^2 \mu^2 + O(1/\sqrt{n}). \tag{24}$$

Comparing with mean aggregation: cost-sensitive weighting improves when

$$\left|\frac{p\, w_+ - q\, w_-}{p\, w_+ + q\, w_-}\right| > \left|\frac{p-q}{p+q}\right|, \tag{25}$$

which holds iff $w_+/w_- > q/p$ (when $q > p$). □

F Multi-Class Extension

The binary CSBM analysis extends to $C > 2$ classes as follows. In a C-class CSBM with intra-class probability p and uniform inter-class probability q:

- The expected fraction of same-class neighbors is $p/(p + (C-1)q) = \mathcal{H}$.
- The between-class scatter matrix $\mathbf{S}_B$ has rank $C - 1$.
- Under mean aggregation, each pairwise class-mean difference is attenuated by $\left(\frac{p-q}{p+(C-1)q}\right)^2$.
- Under cost-sensitive aggregation with $w_+/w_- > q/p$, the same improvement applies to each pairwise difference.

The qualitative conclusion is unchanged: cost-sensitive weighting preserves between-class scatter better than uniform weighting when $w_+/w_- > q/p$.

G Within-Class Scatter

Theorem 2 bounds between-class scatter but not within-class scatter $\mathrm{tr}(\mathbf{S}_W)$. For a node $i \in V_+$:

$$\mathbf{h}_i^{(1)} - \bar{\mathbf{h}}_+^{(1)} = \sum_{j \in \mathcal{N}(i)} w_{ij}(\mathbf{x}_j - \mathbb{E}[\mathbf{x}_j|y_j]) + (\text{edge-composition terms}). \tag{26}$$

The first term involves feature noise averaged over $d_i = \Theta(n)$ neighbors (variance $O(d/n)$). Under cost-sensitive weighting, same-class neighbors receive higher weight, so the aggregated representation is dominated by same-class features. The within-class scatter is bounded by $\mathbb{E}[\mathrm{tr}(\mathbf{S}_W)] \leq n\, d/d_{\text{eff}}$, where $d_{\text{eff}} = \mathbb{E}[\sum_j w_{ij}^2]^{-1}$. A complete formal bound requires tracking the correlation between the random graph and the cost function (which depends on features), making a tight bound technically challenging.

H Reproducibility

Hardware. All experiments on a single machine with an 8-core CPU and 32GB RAM.

Software. Python, PyTorch, PyTorch Geometric. Source code: `https://github.com/eyal-weiss/CSNA-public`.

Splits. 10 random 60/20/20 train/validation/test splits per dataset, generated with seed 42.

Training. Early stopping with patience 50, maximum 300 epochs. Model checkpoint: best validation accuracy. Test evaluation: once, using saved checkpoint.

Tuning protocol. All methods tuned over the same grid:

- Learning rate: $\{0.01, 0.005\}$
- Hidden dimension: $\{64, 128\}$
- CSNA additionally: $\tau \in \{0.1, 0.5, 1.0, 2.0\}$

Best configuration selected by mean validation accuracy over 3 tuning splits. Final results reported on all 10 splits.