ASPECT: NODE-LEVEL ADAPTIVE SPECTRAL FUSION FOR GRAPH CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Spectral graph contrastive learning constructs complementary low- and high-frequency views, but existing methods typically combine them using graph-level or node-agnostic fusion rules. We show that such global fusion can incur irreducible regret when nodes have heterogeneous spectral preferences, motivating node-wise spectral fusion. We propose ASPECT, which learns a node-wise policy to fuse low- and high-frequency views and regularizes this policy using channel-wise contrastive evidence. We further introduce ASPECT-S, a stability-aware extension that incorporates channel-wise sensitivity estimated from generated graph-structure and feature perturbations, with a Rayleigh-based bias for perturbation search. Experiments on homophilic and heterophilic graphs show that ASPECT consistently improves clean representation quality over competitive spectral and graph contrastive baselines, while ASPECT-S further improves performance under joint structural and feature perturbations.

1 INTRODUCTION

Graph contrastive learning (GCL) enables self-supervised representation learning on graphs (Velickovic et al., 2019; You et al., 2020; Zhu et al., 2020b). Real-world graphs often contain heterogeneous local structures: some nodes benefit from smooth, low-frequency information, while others benefit from high-frequency components near heterophilic boundaries (Zhu et al., 2020a; Lim et al., 2021; Zheng et al., 2022a). Spectral GCL methods construct complementary low- and high-frequency views to capture both types of signal (Yang & Mirzasoleiman, 2024; Chen et al., 2024; Wan et al., 2024; Zou et al., 2025; Huang et al., 2024). However, existing spectral GCL methods typically combine low- and high-frequency information using fixed, graph-level, or otherwise node-agnostic rules. Such rules may be restrictive when the preferred spectral mixture varies across nodes.

We study fusion granularity as a policy-selection problem. When node groups have sufficiently separated spectral preferences, we prove that even the best graph-wide coefficient can incur irreducible regret relative to a node-wise oracle. This establishes when a shared fusion rule is structurally inadequate and motivates ASPECT's node-wise fusion policy.

However, learning the node-wise gate from the fused contrastive objective alone can be sensitive to pretraining randomness. In a fixed-split diagnostic reported in Appendix C.3, a gate trained only with the fused contrastive objective yields downstream accuracy standard deviations above 2.5 percentage points across pretraining seeds on both Cora and Squirrel. The fused objective updates the gate through the mixed representation while the spectral filters and encoder evolve jointly, without explicitly comparing the two channels' contrastive quality. ASPECT addresses this by constructing a soft self-supervised target from separate low- and high-frequency contrastive losses to guide each node's gate.

ASPECT's node-wise weight is computed from local channel representations and guided by their contrastive utility. Because these local signals can change under graph-structure and feature perturbations, their sensitivity becomes relevant to the learned fusion policy. ASPECT-S therefore extends the policy target with channel-wise sensitivity estimated from generated perturbations, using a Rayleigh-based bias to seek informative spectral shifts. A perturbation-aware bound motivates this utility–sensitivity tradeoff.

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.

Experiments on homophilic and heterophilic benchmarks show that ASPECT improves clean representation quality over competitive GCL baselines, including spectral methods. ASPECT-S further improves performance under the evaluated joint graph-structure and feature perturbations.

Our contributions are summarized as follows.

(1) **Fusion granularity:** We prove a lower bound on the regret of the best graph-level spectral fusion rule relative to a node-wise oracle under heterogeneous spectral preferences.
(2) **Policy learning from channel-wise evidence:** We construct an explicit self-supervised target from channel-wise contrastive losses to guide a node-wise fusion policy.
(3) **Stability-aware extension:** ASPECT-S augments this policy with channel-wise sensitivity evidence from generated perturbations and uses a Rayleigh-based bias to seek informative spectral changes.

Related work is discussed in Appendix A.

2 Theory: Node-wise Spectral Fusion and Stability-Aware Extension

We provide a compact theoretical analysis to motivate ASPECT and its stability-aware variant, ASPECT-S. The main result shows that graph-level spectral fusion is structurally suboptimal when node-wise spectral preferences are heterogeneous. We then give a perturbation-aware view explaining why, when robustness to local perturbations is desired, generated perturbations can serve as stability-aware probes for estimating channel-wise sensitivity. Finally, we connect the resulting channel-wise utility–sensitivity tradeoff to the auxiliary policy regularizer used to train the node-wise spectral policy. Complete assumptions and proofs are deferred to Appendix B.

2.1 Problem Setup

Let $z_{L,v}, z_{H,v} \in \mathbb{R}^d$ denote the low- and high-frequency embeddings of node v. For a fusion coefficient $m \in [0,1]$, define

$$z_v(m) \triangleq m z_{L,v} + (1-m) z_{H,v}, \tag{1}$$

where larger m corresponds to stronger reliance on the low-frequency channel. Let T denote the randomness in the contrastive objective, such as positive-view and negative-sample selection. The standard contrastive surrogate risk for node v is

$$\mathcal{E}_v(m) \triangleq \mathbb{E}_T\big[\ell(z_v(m); T)\big], \tag{2}$$

where $\ell(\cdot; T)$ is a contrastive surrogate loss.

For the stability analysis, let $\mathcal{Q}_v$ be a set of allowable local perturbations around node v. For $\delta \in \mathcal{Q}_v$, let $z_v^\delta(m)$ denote the fused embedding under perturbation δ. Define the perturbation-aware risk and local sensitivity as

$$R_v^{\mathcal{Q}}(m) \triangleq \sup_{\delta \in \mathcal{Q}_v} \mathbb{E}_T\big[\ell(z_v^\delta(m); T)\big], \qquad S_v(m) \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z_v^\delta(m) - z_v(m)\|_2. \tag{3}$$

2.2 Global Fusion Is Structurally Suboptimal

We first consider graph-level fusion, where a single coefficient $\alpha \in [0,1]$ is shared by all nodes:

$$z_v(\alpha) \triangleq \alpha z_{L,v} + (1-\alpha) z_{H,v}, \qquad R_v(\alpha) \triangleq \mathbb{E}_T[\ell(z_v(\alpha); T)].$$

Define the best global risk and the node-wise oracle risk as

$$R_{\text{global}} \triangleq \min_{\alpha \in [0,1]} \frac{1}{|V|} \sum_{v \in V} R_v(\alpha), \qquad R_{\text{oracle}} \triangleq \frac{1}{|V|} \sum_{v \in V} \min_{\alpha_v \in [0,1]} R_v(\alpha_v), \tag{4}$$

and let $\text{Regret} \triangleq R_{\text{global}} - R_{\text{oracle}}$.

Theorem 2.1 (Irreducible regret of global fusion)**.** *Under a μ-quadratic-growth condition and two separated spectral-preference subpopulations with fractions $p_-, p_+ > 0$, whose optimal fusion sets are contained in $[0, a_-]$ and $[a_+, 1]$, respectively, with $\Delta = a_+ - a_- > 0$, the regret of the best global fusion rule satisfies*

$$\text{Regret} \geq \frac{\mu}{2} \frac{p_- p_+}{p_- + p_+} \Delta^2. \tag{5}$$

Theorem 2.1 establishes a limitation of shared fusion coefficients under the stated preference-separation and growth assumptions. It motivates a node-wise policy, but does not prescribe how to learn one without task labels. In ASPECT, the fused contrastive loss updates the gate through a representation that changes jointly with the filters and encoder; it does not separately assess each channel's contrastive quality. ASPECT therefore constructs a detached, soft target from node-wise channel contrastive losses (Section 3.1), giving the gate an explicit self-supervised signal. The target has an entropy-regularized channel-cost interpretation (Appendix B.9). When performance under perturbations matters, the same target can incorporate channel-wise sensitivity, motivating the optional ASPECT-S extension analyzed next.

2.3 From Standard Risk to Stability-Aware Perturbations

A node-wise policy relies on local spectral evidence. When graph structure or features are perturbed, the channels that supply this evidence can respond differently at each node. The following bound relates perturbation-aware risk to standard contrastive risk and local sensitivity.

Theorem 2.2 (Perturbation-aware risk upper bound)**.** *Assume that for every realization of T, the loss $\ell(\cdot; T)$ is L-Lipschitz with respect to its embedding argument on the compact region visited during training. Then, for every node v and every $m \in [0, 1]$,*

$$R_v^{\mathcal{Q}}(m) \leq \mathcal{E}_v(m) + L S_v(m). \tag{6}$$

Theorem 2.2 implies that minimizing the standard contrastive risk alone may leave the sensitivity term in this upper bound uncontrolled. Consequently, a coefficient that minimizes standard risk can be suboptimal for the perturbation-aware upper bound if its standard-risk advantage is outweighed by a larger sensitivity penalty; a formal statement is given in Appendix B.6.

The upper bound above motivates stability-aware signals, but it does not specify how perturbation sensitivity should be exposed during training. When the test-time local shift is unknown but constrained to $\mathcal{Q}_v$, a worst-case perturbation objective upper-bounds the expected risk under any local-shift distribution supported on $\mathcal{Q}_v$; see Appendix B.7. ASPECT-S provides a practical instantiation of this surrogate with generated graph-structure and feature perturbations, which act as empirical probes of channel-wise sensitivity rather than exact optimizers of the supremum defining $S_v(m)$.

2.4 Channel-Wise Sensitivity Induces a Node-Wise Tradeoff

To connect the sensitivity term to dual-channel spectral fusion, define $d_{L,v} \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z_{L,v}^{\delta} - z_{L,v}\|_2$ and $d_{H,v} \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z_{H,v}^{\delta} - z_{H,v}\|_2$. Assume the perturbed fused embedding uses the same fixed coefficient, $z_v^{\delta}(m) = m z_{L,v}^{\delta} + (1 - m) z_{H,v}^{\delta}$. Then, combining Theorem 2.2 with the triangle inequality gives, for every node v and every fixed $m \in [0, 1]$,

$$R_v^{\mathcal{Q}}(m) \leq \mathcal{E}_v(m) + L\big(m d_{L,v} + (1 - m) d_{H,v}\big). \tag{7}$$

This bound shows that the preferred node-wise mixture can depend on both the standard contrastive risk and the relative local sensitivity of the two channels. It allows either channel to be favored depending on the node and its local perturbation behavior. The fixed-coefficient form also matches the ASPECT-S perturbation-generation step, where the gate is computed on the clean graph and held fixed during the inner perturbation search.

3 Method

We introduce ASPECT, a spectral graph contrastive learning method that adaptively fuses low- and high-frequency views at the node level motivated by the regret lower bound in Section 2. ASPECT

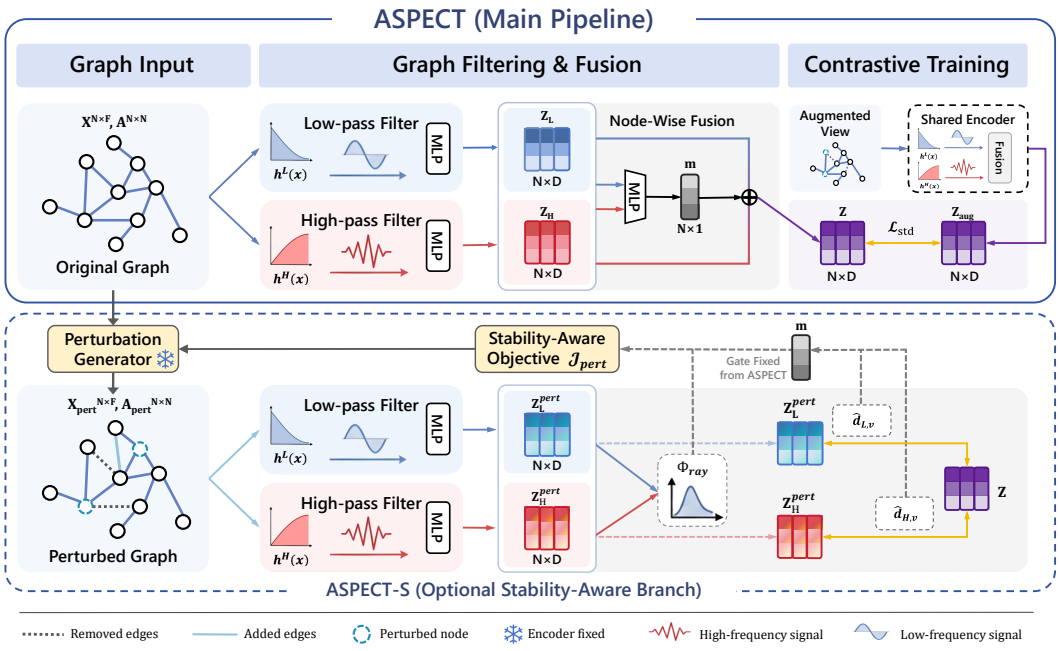

Figure 1: **Overview of ASPECT and ASPECT-S. Top:** ASPECT constructs low- and high-frequency views with learnable spectral filters, learns a node-wise spectral policy m to adaptively fuse the two channels, and optimizes a clean contrastive loss $\mathcal{L}_{\text{std}}$ between the fused embedding $\mathbf{Z}$ and the augmented-view embedding $\mathbf{Z}_{aug}$. **Bottom:** ASPECT-S generates perturbed graphs and features while fixing the clean-graph gate, and uses the resulting perturbed channel representations to obtain empirical channel-wise sensitivity estimates. The Rayleigh-based term Φ_{ray} acts as a spectral search bias for producing informative perturbations. The policy target uses channel-wise contrastive evidence in ASPECT and adds sensitivity evidence in ASPECT-S.

constructs low- and high-frequency views, learns a node-wise spectral policy to fuse them, and optimizes a standard contrastive objective while guiding the policy with channel-wise contrastive evidence. ASPECT-S extends this policy with local channel-sensitivity evidence from generated graph-structure and feature perturbations. During ASPECT-S perturbation generation, the gate is computed on the clean graph and held fixed, matching the fixed-coefficient analysis in Section 2.4; during the outer update, the encoder and gate are updated jointly. Figure 1 summarizes the two modes of policy learning.

3.1 ASPECT: Node-wise Spectral Policy

Low- and high-frequency views. Given a graph $G = (\mathbf{A}, \mathbf{X})$ with normalized Laplacian $\mathbf{L}$, ASPECT builds on learnable spectral filters and focuses on node-wise spectral fusion: how the resulting low- and high-frequency views should be combined for each node. Following spectral GCL practice, we approximate the filters with truncated Chebyshev polynomials:

$$\mathbf{Z}_L = f_\theta\left(\sum_{k=0}^{K} w_k^L T_k(\widetilde{\mathbf{L}})\mathbf{X}\right), \qquad \mathbf{Z}_H = f_\theta\left(\sum_{k=0}^{K} w_k^H T_k(\widetilde{\mathbf{L}})\mathbf{X}\right), \tag{8}$$

where $\widetilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}$, $T_k(\cdot)$ is the k-th Chebyshev basis, f_θ is a shared projection network, and $\{w_k^L, w_k^H\}$ parameterize the two spectral responses. Implementation details for filter parameterization are given in Appendix E.3.

Node-wise spectral policy. For each node v, ASPECT predicts a node-wise spectral policy from the two channel representations:

$$m_v = \sigma(\text{MLP}_g\left([\mathbf{z}_{L,v} \,\|\, \mathbf{z}_{H,v}]\right)), \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function and MLP_g is a lightweight gating network. The fused node representation is

$$\mathbf{z}_v = m_v \mathbf{z}_{L,v} + (1 - m_v)\mathbf{z}_{H,v}. \tag{10}$$

A larger m_v assigns more weight to the low-frequency view, while a smaller m_v assigns more weight to the high-frequency view. This formulation turns spectral fusion from a graph-level scalar into a node-conditioned decision rule, as motivated by Theorem 2.1.

Standard contrastive objective. Let G_{aug} be a randomly augmented view generated by standard graph augmentations such as edge dropping or feature masking, and let $\mathbf{z}_v^{\mathrm{aug}}$ be the fused representation of node v from this view. With normalized InfoNCE loss $\ell_{\mathrm{NCE}}(\cdot,\cdot)$, the standard contrastive objective is

$$\mathcal{L}_{\mathrm{std}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \ell_{\mathrm{NCE}} \left(\mathbf{z}_v, \mathbf{z}_v^{\mathrm{aug}}\right). \tag{11}$$

Channel-wise evidence for policy learning. The fused InfoNCE objective sends gradients to the gate through the final representation in Eq. (10). Because the filters, encoder, and gate are learned jointly, this signal couples their updates without isolating the low- and high-frequency contributions to each node's contrastive quality. ASPECT supplies channel-level evidence for the spectral decision: separate contrastive losses estimate each channel's utility, and a detached target anchors the gate to their relative quality. The channel-wise tradeoff in Eq. (7) also motivates adding sensitivity evidence when perturbations are considered.

For each channel $c \in \{L, H\}$, let $\mathbf{z}_{c,v}$ be the channel-specific representation of node v, and let $\mathbf{z}_{c,v}^{\mathrm{aug}}$ be the corresponding representation from the augmented graph. Define

$$\ell_{c,v}^{\mathrm{std}} = \ell_{\mathrm{NCE}} \left(\mathbf{z}_{c,v}, \mathbf{z}_{c,v}^{\mathrm{aug}}\right), \qquad c \in \{L, H\}. \tag{12}$$

When ASPECT-S is enabled, the generated perturbation gives perturbed channel embeddings $\mathbf{z}_{L,v}^{\mathrm{pert}}$ and $\mathbf{z}_{H,v}^{\mathrm{pert}}$, from which we compute empirical sensitivity evidence:

$$\widehat{d}_{c,v} = \left\|\mathbf{z}_{c,v}^{\mathrm{pert}} - \mathbf{z}_{c,v}\right\|_2, \qquad c \in \{L, H\}. \tag{13}$$

We normalize each evidence type separately over the union of both channels within the current batch: channel-wise losses are normalized over $\{\ell_{L,v}^{\mathrm{std}}, \ell_{H,v}^{\mathrm{std}}\}_{v \in \mathcal{V}}$, and, in ASPECT-S, sensitivity estimates are normalized over $\{\widehat{d}_{L,v}, \widehat{d}_{H,v}\}_{v \in \mathcal{V}}$. This preserves the low/high relative scale within each evidence type. Let $\mathrm{Norm}_\ell(\cdot)$ and $\mathrm{Norm}_d(\cdot)$ denote the corresponding normalizations. The channel cost is

$$b_{c,v} = \begin{cases} \mathrm{Norm}_\ell\left(\ell_{c,v}^{\mathrm{std}}\right), & \text{for ASPECT,} \\ \mathrm{Norm}_\ell\left(\ell_{c,v}^{\mathrm{std}}\right) + \lambda_s \, \mathrm{Norm}_d\left(\widehat{d}_{c,v}\right), & \text{for ASPECT-S,} \end{cases} \qquad c \in \{L, H\}. \tag{14}$$

Thus, ASPECT uses a utility-aware policy target, whereas ASPECT-S uses a utility-sensitivity-aware policy target.

The soft target for the low-frequency policy weight is

$$\widetilde{m}_v = \frac{\exp(-b_{L,v}/\tau_g)}{\exp(-b_{L,v}/\tau_g) + \exp(-b_{H,v}/\tau_g)}, \tag{15}$$

where $\tau_g > 0$ is a gate-temperature hyperparameter. Lower channel cost leads to a larger target weight for that channel. In implementation, $\widetilde{m}_v$ and the channel costs used to construct it are detached from the computational graph, so $\mathcal{L}_{\mathrm{pol}}$ updates the gate through m_v rather than through the pseudo-target. We use the binary cross-entropy form

$$\mathcal{L}_{\mathrm{pol}} = -\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left[\widetilde{m}_v \log m_v + (1 - \widetilde{m}_v) \log(1 - m_v)\right]. \tag{16}$$

This regularizer serves as a soft auxiliary target that aligns the gate with channel-wise evidence while preserving the flexibility of the learned node-wise policy. The ASPECT objective is

$$\mathcal{L}_{\mathrm{ASPECT}} = \mathcal{L}_{\mathrm{std}} + \lambda_{\mathrm{pol}} \mathcal{L}_{\mathrm{pol}}. \tag{17}$$

3.2 ASPECT-S: SENSITIVITY-AWARE POLICY LEARNING

Node-wise fusion depends on local spectral evidence, which can change under graph-structure and feature perturbations. High-frequency representations around local boundaries can be particularly sensitive to these changes. ASPECT-S estimates channel-wise local sensitivity and incorporates it into the same spectral policy. Let $\mathcal{Q}$ denote the allowed local perturbation set over graph structure and node features. ASPECT-S generates a perturbed graph $G_{\text{pert}} = (\mathbf{A}_{\text{pert}}, \mathbf{X}_{\text{pert}})$ by approximately maximizing

$$\mathcal{J}_{\text{pert}} = \mathcal{L}_{\text{gen}} + \lambda_{\text{ray}} \Phi_{\text{ray}} \tag{18}$$

over perturbation variables, with encoder parameters and clean-graph gate values fixed. The gate is fixed only during this inner perturbation-generation step; the encoder and gate are updated jointly in the outer minimization.

For a perturbed graph, define the channel-wise stability loss

$$\mathcal{L}_{\text{stab}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} m_v \ell_{\text{NCE}} \left(\mathbf{z}_{L,v}^{\text{pert}}, \mathbf{z}_v \right) + \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} (1 - m_v) \ell_{\text{NCE}} \left(\mathbf{z}_{H,v}^{\text{pert}}, \mathbf{z}_v \right), \tag{19}$$

where $\mathbf{z}_v$ is the fused clean representation. During perturbation generation, the same expression is used as $\mathcal{L}_{\text{gen}}$ and is maximized over perturbation variables; during the outer update, it is minimized as $\mathcal{L}_{\text{stab}}$ to train the encoder and gate.

The Rayleigh-based term is instantiated as

$$\Phi_{\text{ray}} = \mathcal{R}(\mathbf{A}_{\text{pert}}, \mathbf{Z}_L^{\text{pert}}) - \mathcal{R}(\mathbf{A}_{\text{pert}}, \mathbf{Z}_H^{\text{pert}}), \qquad \mathcal{R}(\mathbf{A}, \mathbf{Z}) = \frac{\text{Tr}(\mathbf{Z}^\top \mathbf{L}_{\mathbf{A}} \mathbf{Z})}{\text{Tr}(\mathbf{Z}^\top \mathbf{Z})}.$$

Since $\mathcal{J}_{\text{pert}}$ is maximized over perturbation variables, Φ_{ray} biases the search toward perturbations that increase the Rayleigh energy of the perturbed low-frequency channel relative to the perturbed high-frequency channel. We use it as a spectral profile-shift bias for generating informative perturbations. A detailed discussion of this term as a spectral search bias is provided in Appendix E.4.

The ASPECT-S objective for the outer update is

$$\mathcal{L}_{\text{ASPECT-S}} = \mathcal{L}_{\text{std}} + \lambda_{\text{stab}} \mathcal{L}_{\text{stab}} + \lambda_{\text{pol}} \mathcal{L}_{\text{pol}}. \tag{20}$$

Here $\mathcal{L}_{\text{pol}}$ uses the ASPECT-S branch of Eq. (14), with empirical sensitivity estimates from Eq. (13). When ASPECT-S is disabled, the method reduces to ASPECT in Eq. (17).

3.3 OPTIMIZATION AND COMPLEXITY

To reduce computational overhead, our implementation uses sampled contrastive scoring to avoid dense pairwise contrastive matrices on large graphs. ASPECT-S further uses four engineering optimizations to reduce the cost of perturbation generation. First, structural perturbations are restricted to sparse candidate edge sets, while feature perturbations are constrained by a prescribed feature budget. Second, we warm up the ASPECT objective and generate perturbations only every few epochs. Third, the policy target $\widetilde{m}_v$ is detached and channel-wise scores are computed without gradient tracking, so $\mathcal{L}_{\text{pol}}$ introduces no additional encoder forward pass. Finally, the Rayleigh term is computed via sparse Laplacian-vector products rather than eigendecomposition. We discuss the incremental computational complexity of ASPECT and ASPECT-S in Section 4.4, with implementation details in Appendix E.

4 EXPERIMENTS

This section empirically evaluates ASPECT and ASPECT-S. Our experiments are organized around four questions: (Q1) Clean representation quality: does node-level adaptive spectral fusion improve performance on homophilic and heterophilic graphs? (Q2) Perturbation performance: does ASPECT-S improve performance under joint graph-structure and feature perturbations when enabled? (Q3) Component contribution: how do the node-wise spectral policy, policy regularizer, Rayleigh-based search bias, and sensitivity-aware target affect performance? (Q4) Policy behavior and complexity: does the learned policy exhibit node-adaptive spectral mixtures, and what incremental computational complexity is introduced?

4.1 EXPERIMENTAL SETUP

Datasets. We conduct node classification experiments on nine widely used benchmark graphs covering both homophilic and heterophilic regimes. Homophilic datasets include `Cora`, `Citeseer`, and `Pubmed` (Sen et al., 2008). Heterophilic datasets include `Cornell`, `Texas`, `Wisconsin`, `Actor`, `Chameleon`, and `Squirrel` (Pei et al., 2020; Rozemberczki et al., 2021). Dataset descriptions and preprocessing details are provided in Appendix D.

Baselines. We compare ASPECT with representative graph representation learning baselines, including general graph contrastive learning methods, augmentation-robust GCL methods, and spectral or heterophily-oriented GCL methods. Detailed descriptions and configurations are provided in Appendix E. Among these methods, PolyGCL (Chen et al., 2024) is the most direct external comparison for our fusion-motivation theory: it constructs dual spectral channels but uses a node-agnostic fusion rule.

Self-supervised training and linear evaluation. Following the standard protocol of Velickovic et al. (2019), each method is first pretrained in a self-supervised manner on the unlabeled graph. We then freeze the encoder and train a linear classifier on top of the learned node representations. We use 10 random data splits with 60%/20%/20% train/validation/test partitions following Chien et al. (2020), and report mean test accuracy with standard deviation across splits. Hyperparameters are selected using validation accuracy on the clean graph only, so that perturbation results are not tuned on perturbed data.

Perturbation evaluation protocol. To evaluate performance under graph-structure and feature perturbations, we evaluate a joint perturbation setting that combines graph-structure and feature perturbations. Following Feng et al. (2024), Metattack (Zügner & Günnemann, 2019) is used to perturb graph edges, and random feature masking is applied to corrupt node attributes. In the fixed-budget setting, the edge perturbation ratio and feature masking ratio are both set to 10%. In the variable-budget setting, both ratios are swept jointly using the same budget. For each split, Metattack uses only training labels and a fixed surrogate model; full perturbation-generation details are provided in Appendix E.2. Datasets with very small node counts are omitted from perturbation evaluation due to high variance and unstable graph statistics under perturbations; the evaluated datasets are explicitly stated in each table or figure.

4.2 CLEAN REPRESENTATION QUALITY

Table 1 reports clean linear-probe performance; the full comparison is in Appendix C.4. ASPECT ranks first on 8 of 9 datasets and second on the remaining one. It exceeds PolyGCL, the closest node-agnostic dual-spectral baseline, on all nine. The matched Global Fusion ablation in Table 3 isolates fusion granularity: ASPECT gains 1.45, 0.82, 1.03, and 2.34 points on Cora, Pubmed, Actor, and Squirrel, respectively. ASPECT-S retains strong clean performance, ranking first on Squirrel and second on most other datasets.

Large-graph evaluation. On arXiv-year, ASPECT achieves $43.65 \pm 0.32\%$ accuracy, compared with $43.09 \pm 0.29\%$ for PolyGCL and $41.80 \pm 0.21\%$ for the matched Global Fusion ablation. The experimental setup, accuracy, epoch time, and peak GPU memory for all methods are reported in Appendix C.1 (Table 4).

4.3 PERFORMANCE UNDER PERTURBATIONS

We evaluate a fixed-budget joint perturbation setting, where graph-structure and feature perturbations are applied simultaneously. Metattack perturbs 10% of edges and feature masking randomly masks 10% of entries in the node-feature matrix. Table 2 summarizes results against representative baselines; the full comparison is provided in Appendix C.5. Across the full comparison, ASPECT-S achieves the highest perturbed accuracy on all evaluated datasets and the lowest average relative accuracy drop (**6.51%**). Compared with ASPECT, ASPECT-S consistently improves perturbed accuracy, showing that stability-aware training provides useful training signals when robustness to graph-structure and feature perturbations is desired. Compared with PolyGCL, the closest node-agnostic dual-spectral

Table 1: Node classification accuracy (mean ± standard deviation, %) on nine homophilic and heterophilic benchmarks under the linear evaluation protocol. The main text reports representative baselines; full results are provided in Appendix C.4. Boldface indicates the best performance and underline indicates the second-best performance.

Methods	Homophilic Datasets			Heterophilic Datasets					
	Cora	Citeseer	Pubmed	Cornell	Texas	Wisconsin	Actor	Chameleon	Squirrel
DGI	85.88 ± 0.95	76.44 ± 0.84	82.13 ± 0.24	70.82 ± 2.71	81.48 ± 2.79	75.00 ± 4.22	32.09 ± 1.18	58.23 ± 0.70	38.80 ± 0.76
GRACE	83.27 ± 0.74	73.79 ± 0.57	81.71 ± 0.14	60.66 ± 2.94	75.74 ± 3.12	72.13 ± 1.99	31.97 ± 1.13	59.52 ± 2.65	42.68 ± 1.10
GCA	84.09 ± 0.85	75.23 ± 1.19	82.01 ± 0.34	53.11 ± 4.01	81.97 ± 1.58	73.50 ± 2.85	31.13 ± 1.11	65.54 ± 1.10	47.13 ± 0.93
GREET	85.16 ± 0.77	79.06 ± 1.34	85.64 ± 0.28	78.36 ± 2.77	78.03 ± 3.94	84.63 ± 2.10	37.12 ± 0.67	60.57 ± 1.03	42.80 ± 1.01
CCA-SSG	87.39 ± 0.89	79.60 ± 1.01	84.95 ± 0.26	78.69 ± 4.61	87.87 ± 1.89	82.88 ± 3.58	34.86 ± 1.13	59.84 ± 1.21	41.50 ± 1.12
SP-GCL	82.99 ± 1.18	75.54 ± 1.06	85.74 ± 0.21	69.41 ± 1.49	69.76 ± 1.23	69.34 ± 0.77	35.92 ± 0.67	69.23 ± 1.23	53.05 ± 1.05
PolyGCL	87.57 ± 0.62	79.81 ± 0.85	87.15 ± 0.27	82.62 ± 3.11	88.03 ± 1.80	85.50 ± 1.88	41.15 ± 0.88	71.62 ± 0.96	56.49 ± 0.72
S3GCL	87.04 ± 1.25	77.48 ± 0.80	86.03 ± 0.37	81.27 ± 3.67	86.12 ± 3.91	84.56 ± 2.71	40.06 ± 1.58	71.88 ± 1.91	56.90 ± 1.37
RDGI	83.53 ± 1.23	78.99 ± 0.80	80.89 ± 1.55	67.21 ± 6.06	69.01 ± 4.59	56.75 ± 4.12	32.74 ± 1.27	59.95 ± 1.11	42.71 ± 0.70
ARIEL	87.30 ± 0.71	79.53 ± 0.61	86.42 ± 0.47	70.70 ± 2.46	76.19 ± 5.02	71.15 ± 2.38	37.68 ± 1.03	64.53 ± 1.47	42.42 ± 1.53
ASPECT	**88.94** ± 1.10	**81.82** ± 0.90	**87.75** ± 1.03	**89.18** ± 2.77	**91.76** ± 2.09	**90.03** ± 2.09	**42.64** ± 1.55	**72.88** ± 1.90	59.39 ± 1.76
ASPECT-S	88.69 ± 0.72	81.30 ± 0.85	86.92 ± 0.69	88.68 ± 2.26	90.41 ± 1.97	88.00 ± 2.12	41.73 ± 1.30	72.48 ± 1.43	**59.91** ± 0.90

fusion baseline, ASPECT-S reduces the average drop (6.51% vs. 13.22%), suggesting that node-level adaptive fusion can work effectively together with stability-aware perturbation signals under perturbed evaluation. ASPECT-S also outperforms ARIEL, a robust GCL baseline, in both average degradation and absolute perturbed accuracy. Variable-budget results are provided in Appendix C.5, where edge perturbation and feature masking ratios are swept jointly using the same budget ratio.

Table 2: Node classification accuracy (mean ± standard deviation, %) under fixed-budget joint graph-structure and feature perturbations. We report selected representative baselines in the main text and provide the full comparison in Appendix C.5. Boldface indicates the best result and underline indicates the second-best result.

Methods	Homophilic Datasets			Heterophilic Datasets			Avg. Drop (%)
	Cora	Citeseer	Pubmed	Actor	Chameleon	Squirrel	
DGI	79.62 ± 0.62	72.25 ± 0.85	74.29 ± 1.01	30.28 ± 1.32	51.47 ± 0.70	32.94 ± 0.73	9.11
GRACE	77.08 ± 1.28	70.67 ± 0.86	75.25 ± 0.60	30.78 ± 0.71	51.38 ± 1.75	32.76 ± 1.07	10.03
GCA	76.39 ± 0.92	56.55 ± 1.31	71.32 ± 0.87	31.87 ± 0.97	58.75 ± 1.09	37.20 ± 0.90	12.68
GREET	78.80 ± 1.45	75.44 ± 0.59	79.47 ± 0.57	34.46 ± 1.23	51.77 ± 1.55	35.64 ± 1.32	9.61
CCA-SSG	82.79 ± 1.28	74.88 ± 0.72	77.01 ± 0.90	30.70 ± 0.77	49.63 ± 1.09	31.23 ± 1.44	12.57
SP-GCL	76.32 ± 1.11	70.12 ± 1.07	74.76 ± 0.79	30.77 ± 0.76	62.02 ± 1.72	41.94 ± 1.32	12.29
PolyGCL	83.18 ± 0.78	72.51 ± 1.25	77.82 ± 0.83	37.35 ± 0.90	59.01 ± 1.35	40.89 ± 1.40	13.22
S3GCL	80.31 ± 0.62	71.72 ± 1.40	79.46 ± 1.57	36.03 ± 1.28	59.89 ± 1.99	40.29 ± 1.75	13.12
RDGI	78.85 ± 0.96	73.92 ± 0.68	74.12 ± 1.41	30.37 ± 1.47	52.66 ± 0.94	34.00 ± 0.63	10.03
ARIEL	84.80 ± 1.01	76.17 ± 1.39	81.08 ± 0.95	32.33 ± 0.43	56.18 ± 1.08	36.09 ± 1.11	9.22
ASPECT	84.08 ± 1.41	77.44 ± 0.86	82.38 ± 0.52	36.14 ± 0.47	64.18 ± 1.93	46.37 ± 1.08	11.01
ASPECT-S	**85.30** ± 0.72	**78.86** ± 0.70	**84.93** ± 0.44	**39.20** ± 0.63	**66.97** ± 1.70	**50.17** ± 0.92	**6.51**

4.4 Ablation, Policy Behavior, and Complexity

Component ablation. Table 3 tests the main components on representative homophilic and heterophilic datasets. The matched Global Fusion variant uses the same dual-channel encoder and contrastive objective. A node-wise gate trained without PolicyReg improves over Global Fusion on Actor and Squirrel, but falls below it on Cora and Pubmed: finer-grained fusion alone gives inconsistent gains. ASPECT exceeds both variants on all four datasets, supporting the benefit of channel-wise contrastive evidence. The fixed-split diagnostic in Appendix C.3 shows lower pretraining-seed sensitivity with PolicyReg. For ASPECT-S, we ablate two components of stability-aware training. Removing the Rayleigh-based spectral search bias weakens performance under joint graph-structure

and feature perturbations, suggesting that spectrally informative perturbation generation is beneficial. Removing the sensitivity term from the policy target causes a larger degradation, especially under perturbations, supporting the usefulness of empirical channel-wise sensitivity estimates in the stability-aware extension.

Hyperparameter sensitivity. One-at-a-time sweeps on Cora and Squirrel show less than one point of accuracy variation across the tested gate-temperature range. Setting $\lambda_s = 0$ reduces perturbed accuracy by 2.49/3.18 points on Cora/Squirrel, while setting $\lambda_{\text{ray}} = 0$ reduces it by 1.62/2.02 points. The settings and complete results are in Appendix C.2 (Tables 5 and 6).

Table 3: Ablation study on representative homophilic and heterophilic datasets. We report node classification accuracy (mean ± standard deviation, %) on clean graphs and under fixed-budget joint graph-structure and feature perturbations. Bold indicates the best performance within each ASPECT or ASPECT-S block.

Variant	Cora		Pubmed		Actor		Squirrel	
	Clean	Pert.	Clean	Pert.	Clean	Pert.	Clean	Pert.
ASPECT	**88.94** ± 1.10	**84.08** ± 1.41	**87.75** ± 1.03	**82.38** ± 0.52	**42.64** ± 1.55	36.14 ± 0.47	**59.39** ± 1.76	**46.37** ± 1.08
Global Fusion	87.49 ± 0.74	83.20 ± 0.85	86.93 ± 0.68	78.96 ± 0.97	41.61 ± 0.99	**37.07** ± 1.51	57.05 ± 0.88	43.39 ± 0.94
w/o PolicyReg	87.15 ± 1.76	81.19 ± 1.46	85.40 ± 1.90	79.17 ± 1.67	41.76 ± 2.08	35.28 ± 2.10	57.92 ± 1.00	41.90 ± 1.49
ASPECT-S	**88.69** ± 0.72	**85.30** ± 0.72	**86.92** ± 0.69	**84.93** ± 0.44	**41.73** ± 1.30	**39.20** ± 0.63	**59.91** ± 0.90	**50.17** ± 0.92
w/o Rayleigh	87.91 ± 0.99	83.64 ± 1.04	86.18 ± 1.06	82.83 ± 0.83	41.22 ± 1.43	38.06 ± 1.36	58.43 ± 1.77	47.61 ± 1.87
w/o SensPolicy	87.09 ± 1.95	83.01 ± 1.47	85.49 ± 1.00	81.98 ± 1.32	41.13 ± 0.98	37.49 ± 0.86	57.06 ± 2.41	46.93 ± 1.95

Policy behavior. Figure 2 shows that the learned policy is both node-adaptive and aligned with channel-wise evidence. The weights m_v do not collapse to a global coefficient: `Cora` is biased toward larger low-frequency weights, while `Squirrel` has a broader distribution, suggesting stronger node-wise heterogeneity. Moreover, the average m_v increases with $b_{H,v} - b_{L,v}$, indicating that the policy assigns larger low-frequency weights when the low-frequency channel has lower estimated cost.

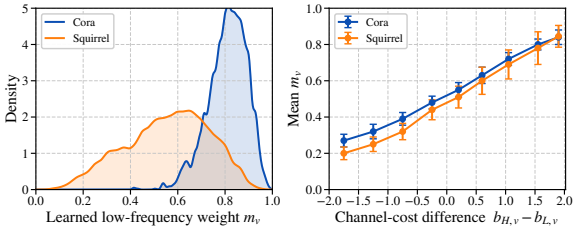

Figure 2: Policy behavior of ASPECT on clean graphs. Left: m_v distributions. Right: binned m_v versus $b_{H,v} - b_{L,v}$. Error bars indicate within-bin variability.

Complexity discussion. ASPECT's sparse spectral propagation costs $O(Kmd)$, while the shared gate is node-linear and needs no additional encoder pass. ASPECT-S adds periodic, sparse perturbation generation; its detailed time and memory costs are in Appendix E.6.

5 Conclusion

ASPECT addresses two coupled problems in spectral graph contrastive learning: the granularity of low-/high-frequency fusion and the self-supervised learning of a node-wise fusion policy. A regret lower bound characterizes the cost of global fusion under heterogeneous node preferences, while channel-wise contrastive evidence guides each node's spectral decision. ASPECT-S extends the same policy with local channel-sensitivity evidence from generated perturbations. Across homophilic and heterophilic benchmarks, ASPECT improves clean representation quality, and ASPECT-S improves performance under the evaluated joint perturbations. Limitations and future directions are discussed in Appendix F.

AI use statement

Generative AI tools were used to assist with experimental code development, processing experimental results, language editing, and migration of the manuscript to the ICLR template. All experimental

results, technical claims, and final manuscript content were reviewed and verified by the authors, who take full responsibility for the work.

Ethics statement

This study uses existing, publicly available graph benchmarks and does not involve recruiting human participants or collecting new personal data. We do not release new individual-level data. Any application of the method to sensitive relational data would require a separate assessment of privacy, fairness, and the applicable data-use conditions.

Reproducibility statement

The experimental protocol, dataset sources, preprocessing, and data splits are described in Appendix D; model configurations, hyperparameters, and implementation details are in Appendix E. The additional arXiv-year and hyperparameter-sensitivity settings and results are in Appendix C, and the assumptions and proofs are in Appendix B. We will provide an anonymized code package, including an environment specification, with the submission.

References

Piotr Bielak, Tomasz Kajdanowicz, and Nitesh V Chawla. Graph barlow twins: A self-supervised representation learning framework for graphs. *Knowledge-Based Systems*, 256:109631, 2022.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2021.

Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. In *Advances in Neural Information Processing Systems*, 2019.

Jingyu Chen, Runlin Lei, and Zhewei Wei. PolyGCL: GRAPH CONTRASTIVE LEARNING via learnable spectral polynomial filters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y21ZO6M86t.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. Ariel: Adversarial graph contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–22, 2024.

Haoyu Han, Juanhui Li, Wei Huang, Xianfeng Tang, Hanqing Lu, Chen Luo, Hui Liu, and Jiliang Tang. Node-wise filtering in graph neural networks: A mixture of experts approach, 2024.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pp. 4116–4126. PMLR, 2020.

Dongxiao He, Chundong Liang, Huixin Liu, Mingxiang Wen, Pengfei Jiao, and Zhiyong Feng. Block modeling-guided graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4022–4029, 2022.

Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 594–604. ACM, 2022. doi: 10.1145/3534678.3539321.

Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. Graphmae2: A decoding-enhanced masked self-supervised graph learner. In *Proceedings of the ACM Web Conference 2023*, pp. 737–746. ACM, 2023. doi: 10.1145/3543507.3583379.

Rui Huang, Ping Li, and Kai Zhang. Dpgcl: Dual pass filtering based graph contrastive learning. *Neural Networks*, 179:106517, 2024.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in neural information processing systems*, 33:16199–16210, 2020.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in neural information processing systems*, 33:2983–2994, 2020.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. *Advances in neural information processing systems*, 34:20887–20902, 2021.

Lu Lin, Ethan Blaser, and Hongning Wang. Graph structural attack by perturbing spectral distance. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 989–998, 2022.

Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent CS Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 4516–4524, 2023.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Revisiting heterophily for graph neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1362–1375, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pp. 259–270, 2020.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1150–1160. ACM, 2020. doi: 10.1145/3394486.3403168.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Chengxi Song, Lingfeng Niu, and Minglong Lei. Two-level adversarial attacks for graph neural networks. *Information Sciences*, 654:119877, 2024.

Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:15920–15933, 2021.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (poster)*, 2(3):4, 2019.

Guancheng Wan, Yijun Tian, Wenke Huang, Nitesh V Chawla, and Mang Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *Forty-first International Conference on Machine Learning*, 2024.

Haonan Wang, Jieyu Zhang, Qi Zhu, Wei Huang, Kenji Kawaguchi, and Xiaokui Xiao. Single-pass contrastive learning can work for both homophilic and heterophilic graph. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=244KePn09i.

Jiarong Xu, Yang Yang, Junru Chen, Xin Jiang, Chunping Wang, Jiangang Lu, and Yizhou Sun. Unsupervised adversarially robust representation learning on graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4290–4298, 2022.

Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019.

Wenhan Yang and Baharan Mirzasoleiman. Graph contrastive learning under heterophily via graph filters. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, UAI '24. JMLR.org, 2024.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34:76–89, 2021.

Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. In *Advances in Neural Information Processing Systems*, 2020.

Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey. *arXiv preprint arXiv:2202.07082*, 2022a.

Yizhen Zheng, Shirui Pan, Vincent Lee, Yu Zheng, and Philip S Yu. Rethinking and scaling up graph contrastive learning: An extremely efficient approach with group discrimination. *Advances in Neural Information Processing Systems*, 35:10809–10820, 2022b.

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.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020a.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020b.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, pp. 2069–2080, 2021.

Ziyun Zou, Yinghui Jiang, Lian Shen, Juan Liu, and Xiangrong Liu. Loha: Direct graph spectral contrastive learning between low-pass and high-pass views. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13492–13500, 2025.

Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2847–2856. ACM, 2018. doi: 10.1145/3219819.3220078.

Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bylnx209YX.

A Related Work

A.1 Self-Supervised Graph Representation Learning

Self-supervised graph representation learning has been extensively studied to mitigate label scarcity on graphs. Early approaches largely follow mutual-information maximization and contrastive paradigms, such as DGI (Velickovic et al., 2019), MVGRL (Hassani & Khasahmadi, 2020), and GMI (Peng et al., 2020). Subsequent methods emphasize augmentation-driven contrastive objectives, including GraphCL (You et al., 2020), GRACE (Zhu et al., 2020b), and adaptive augmentation in GCA (Zhu et al., 2021), while other works improve scalability and efficiency via alternative discrimination schemes (Zheng et al., 2022b). Beyond contrastive learning, non-contrastive objectives based on bootstrapping or redundancy reduction, such as BGRL (Thakoor et al., 2021), Graph Barlow Twins (Bielak et al., 2022), and CCA-SSG (Zhang et al., 2021), alleviate the reliance on negative samples and manually designed negative pairs.

Generative pretext tasks have also been widely explored on graphs. Masked graph autoencoders, such as GraphMAE (Hou et al., 2022) and GraphMAE2 (Hou et al., 2023), reconstruct masked node features and demonstrate strong representation quality. Cross-graph pretraining frameworks such as GCC (Qiu et al., 2020) learn transferable structural patterns via subgraph-level instance discrimination. These methods provide strong general graph SSL baselines. Our work is complementary: rather than introducing a new pretext task, ASPECT focuses on how low- and high-frequency graph views should be adaptively fused when node-wise spectral preferences are heterogeneous.

A.2 Heterophily, Mixed Graphs, and Frequency-Aware Learning

A key challenge in graph learning is heterophily, where adjacent nodes may have dissimilar labels or features. Classical message-passing GNNs can degrade under heterophily due to over-smoothing and the low-pass nature of neighborhood aggregation (Zhu et al., 2020a; Lim et al., 2021). Recent surveys summarize this line of work and categorize architectural remedies for heterophilous graphs (Zheng et al., 2022a). Representative heterophily-oriented methods exploit structural patterns beyond immediate neighborhoods, such as block modeling guidance (He et al., 2022), or explicitly strengthen heterophily discrimination, such as GREET (Liu et al., 2023).

From a graph signal processing perspective, homophilic and heterophilic regions can benefit from different spectral components. Frequency-adaptive GNNs, such as FAGCN (Bo et al., 2021), ACM-GCN (Luan et al., 2022), and Node-MoE (Han et al., 2024), adapt frequency responses or channel mixtures to local structure in supervised node-classification settings. In self-supervised learning, spectral or frequency-aware contrastive methods construct complementary views using polynomial spectral filters in PolyGCL (Chen et al., 2024), hybrid spectral-spatial pipelines in S3GCL (Wan et al., 2024), and heterophily-aware dual filtering in HLCL (Yang & Mirzasoleiman, 2024). More recent methods further explore explicit low-/high-pass view contrast (Zou et al., 2025) or multi-pass filtering designs (Huang et al., 2024). These lines motivate two linked questions in spectral GCL: how to fuse low-/high-frequency contrastive views at node level, and how to learn the resulting policy without task-label supervision. ASPECT addresses them with a node-wise spectral policy guided by channel-wise contrastive evidence.

A.3 Graph Perturbations and Robust Graph SSL

Graph neural networks can be sensitive to perturbations of graph structure and node features. Classic targeted attacks include Nettack (Zügner et al., 2018), while meta-learning-based poisoning attacks such as Metattack (Zügner & Günnemann, 2019) perturb graph structure to degrade downstream performance. Further studies analyze graph attacks and defenses from optimization and topology perspectives (Xu et al., 2019), and propose additional perturbation objectives, including spectral-distance-driven perturbations (Lin et al., 2022) and multi-level attack strategies (Song et al., 2024). Robust graph learning methods include robust GCN variants (Zhu et al., 2019), graph structure learning for denoising (Jin et al., 2020), and edge reweighting or pruning mechanisms such as GNNGuard (Zhang & Zitnik, 2020). Certified robustness methods such as Graph-Cert (Bojchevski & Günnemann, 2019) provide worst-case guarantees under graph perturbations for certain model classes.

Robustness has also been studied in self-supervised and contrastive learning. General adversarial contrastive learning principles (Kim et al., 2020; Ho & Nvasconcelos, 2020; Jiang et al., 2020; Madry et al., 2017) have inspired graph adaptations. In graph SSL, adversarial augmentation and robust objectives have been explored in AD-GCL (Suresh et al., 2021), RDGI (Xu et al., 2022), and ARIEL (Feng et al., 2024). These works primarily aim to improve robustness through perturbation-aware training objectives. ASPECT-S uses generated graph-structure and feature perturbations to estimate channel-specific sensitivity for node-wise spectral fusion.

A.4 Positioning of ASPECT

ASPECT is positioned at the intersection of spectral graph contrastive learning and node-adaptive representation learning. Existing spectral GCL methods show the value of constructing low- and high-frequency views, but their fusion is often graph-level or node-agnostic. Our theory shows that such global fusion can be structurally suboptimal on mixed graphs with heterogeneous node-wise spectral preferences. Supervised node-adaptive filtering establishes the value of local frequency choices; ASPECT learns those choices for low-/high-frequency contrastive views using channel-wise self-supervised evidence. This connects fusion granularity to policy learning within spectral GCL.

ASPECT-S extends the policy with local channel-sensitivity evidence. Generated graph-structure and feature perturbations estimate how each channel responds around a node, while a Rayleigh-based bias encourages informative spectral changes during perturbation search. The resulting policy jointly considers contrastive utility and local channel sensitivity.

B Detailed Theory

This appendix provides the formal assumptions and proofs for Section 2. The analysis is conducted at the representation/fusion level. Its scope is to justify node-wise spectral fusion and motivate the stability-aware extension ASPECT-S. It treats channel sensitivity as node- and perturbation-dependent, and analyzes the structural limitation of graph-level fusion.

B.1 Notation and Scope

For each node v, let $z_{L,v}, z_{H,v} \in \mathbb{R}^d$ denote the low- and high-frequency embeddings produced by the encoder. For a fusion coefficient $m \in [0, 1]$, define

$$z_v(m) \triangleq m z_{L,v} + (1 - m) z_{H,v}.$$

Larger m corresponds to stronger reliance on the low-frequency channel. In the model, the learned gate outputs a node-wise coefficient m_v from the spectral views; in the analysis below, m is treated as fixed when studying a single node.

Let T denote the randomness in the contrastive objective, such as positive-view and negative-sample selection. The standard contrastive surrogate risk for node v is

$$\mathcal{E}_v(m) \triangleq \mathbb{E}_T[\ell(z_v(m); T)].$$

For stability analysis, let $\mathcal{Q}_v$ be a set of allowable local graph-structure and/or feature perturbations around node v. For $\delta \in \mathcal{Q}_v$, let $z_v^\delta(m)$ denote the fused representation under perturbation δ. The perturbation-aware risk and sensitivity are

$$R_v^{\mathcal{Q}}(m) \triangleq \sup_{\delta \in \mathcal{Q}_v} \mathbb{E}_T[\ell(z_v^\delta(m); T)], \qquad S_v(m) \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z_v^\delta(m) - z_v(m)\|_2.$$

The main result shows that graph-level fusion is structurally limited under heterogeneous node-wise spectral preferences. The stability analysis then motivates ASPECT-S when robustness to local perturbations is desired.

B.2 Formal Assumptions for Global Fusion Regret

We first formalize the assumptions used in Theorem 2.1. Consider graph-level fusion with a single coefficient $\alpha \in [0, 1]$ shared by all nodes:

$$z_v(\alpha) \triangleq \alpha z_{L,v} + (1 - \alpha) z_{H,v}, \qquad R_v(\alpha) \triangleq \mathbb{E}_T[\ell(z_v(\alpha); T)].$$

Assumption B.1 (Continuity and minimizer existence)**.** For every node v, the function $R_v(\alpha)$ is continuous on the compact interval $[0,1]$. Therefore, the optimal fusion set

$$A_v^\star \triangleq \arg\min_{\alpha\in[0,1]} R_v(\alpha)$$

is nonempty and compact. Let

$$R_v^\star \triangleq \min_{\alpha\in[0,1]} R_v(\alpha).$$

Assumption B.2 (Quadratic growth / error bound)**.** There exists $\mu > 0$ such that for every node v and every $\alpha \in [0,1]$,

$$R_v(\alpha) - R_v^\star \geq \frac{\mu}{2}\,\mathrm{dist}^2(\alpha, A_v^\star),$$

where

$$\mathrm{dist}(\alpha, A_v^\star) \triangleq \inf_{\beta\in A_v^\star} |\alpha - \beta|.$$

Assumption B.3 (Separated node-wise spectral-preference subpopulations)**.** There exist two disjoint node subsets $V^-, V^+ \subseteq V$, not necessarily covering all nodes, with

$$p_- \triangleq \frac{|V^-|}{|V|} > 0, \qquad p_+ \triangleq \frac{|V^+|}{|V|} > 0,$$

and constants $0 \leq a_- < a_+ \leq 1$ such that

$$A_v^\star \subseteq [0, a_-], \quad \forall v \in V^-, \qquad A_v^\star \subseteq [a_+, 1], \quad \forall v \in V^+.$$

Let $\Delta \triangleq a_+ - a_- > 0$.

Assumption B.3 only requires two separated subpopulations with nonzero mass. It does not require all nodes to belong to $V^- \cup V^+$; nodes outside these subsets are allowed to have arbitrary optimal fusion sets.

B.3 Proof of Theorem 2.1

Proof. Recall the definitions

$$R_{\mathrm{global}} \triangleq \min_{\alpha\in[0,1]} \frac{1}{|V|}\sum_{v\in V} R_v(\alpha), \qquad R_{\mathrm{oracle}} \triangleq \frac{1}{|V|}\sum_{v\in V}\min_{\alpha_v\in[0,1]} R_v(\alpha_v),$$

and

$$\mathrm{Regret} \triangleq R_{\mathrm{global}} - R_{\mathrm{oracle}}.$$

Since

$$R_{\mathrm{oracle}} = \frac{1}{|V|}\sum_{v\in V} R_v^\star,$$

we have

$$\mathrm{Regret} = \min_{\alpha\in[0,1]} \frac{1}{|V|}\sum_{v\in V}\big(R_v(\alpha) - R_v^\star\big).$$

By Assumption B.2,

$$\mathrm{Regret} \geq \frac{\mu}{2}\min_{\alpha\in[0,1]} \frac{1}{|V|}\sum_{v\in V}\mathrm{dist}^2(\alpha, A_v^\star).$$

For $v \in V^-$, Assumption B.3 gives $A_v^\star \subseteq [0, a_-]$. Therefore,

$$\mathrm{dist}(\alpha, A_v^\star) \geq (\alpha - a_-)_+,$$

where $(x)_+ \triangleq \max\{x, 0\}$. Similarly, for $v \in V^+$, $A_v^\star \subseteq [a_+, 1]$, and hence

$$\mathrm{dist}(\alpha, A_v^\star) \geq (a_+ - \alpha)_+.$$

All nodes outside $V^- \cup V^+$ contribute nonnegative terms, so dropping them preserves a valid lower bound:

$$\mathrm{Regret} \geq \frac{\mu}{2}\min_{\alpha\in[0,1]}\left[p_-(\alpha - a_-)_+^2 + p_+(a_+ - \alpha)_+^2\right].$$

It remains to solve the one-dimensional minimization problem

$$\min_{\alpha\in[0,1]} \left[p_-(\alpha - a_-)_+^2 + p_+(a_+ - \alpha)_+^2\right].$$

First consider $\alpha \in [a_-, a_+]$. The objective becomes

$$g(\alpha) = p_-(\alpha - a_-)^2 + p_+(a_+ - \alpha)^2.$$

The derivative is

$$g'(\alpha) = 2p_-(\alpha - a_-) - 2p_+(a_+ - \alpha).$$

Setting $g'(\alpha) = 0$ gives

$$\alpha^\star = \frac{p_- a_- + p_+ a_+}{p_- + p_+}.$$

Since $a_- < a_+$ and $p_-, p_+ > 0$, this minimizer lies in $[a_-, a_+]$. Moreover,

$$\alpha^\star - a_- = \frac{p_+}{p_- + p_+}\Delta, \qquad a_+ - \alpha^\star = \frac{p_-}{p_- + p_+}\Delta.$$

Thus,

$$g(\alpha^\star) = p_- \left(\frac{p_+}{p_- + p_+}\Delta\right)^2 + p_+ \left(\frac{p_-}{p_- + p_+}\Delta\right)^2 = \frac{p_- p_+}{p_- + p_+}\Delta^2.$$

Now consider $\alpha < a_-$. Then

$$p_-(\alpha - a_-)_+^2 + p_+(a_+ - \alpha)_+^2 = p_+(a_+ - \alpha)^2 \geq p_+\Delta^2.$$

Since

$$p_+\Delta^2 \geq \frac{p_- p_+}{p_- + p_+}\Delta^2,$$

the lower bound is no smaller outside the interval on the left. Similarly, if $\alpha > a_+$, then

$$p_-(\alpha - a_-)_+^2 + p_+(a_+ - \alpha)_+^2 = p_-(\alpha - a_-)^2 \geq p_-\Delta^2 \geq \frac{p_- p_+}{p_- + p_+}\Delta^2.$$

Therefore,

$$\min_{\alpha\in[0,1]} \left[p_-(\alpha - a_-)_+^2 + p_+(a_+ - \alpha)_+^2\right] = \frac{p_- p_+}{p_- + p_+}\Delta^2.$$

Combining this with the previous regret lower bound yields

$$\text{Regret} \geq \frac{\mu}{2}\frac{p_- p_+}{p_- + p_+}\Delta^2.$$

When $V^- \cup V^+ = V$, let $r = |V^+|/|V|$. Then $p_+ = r$ and $p_- = 1 - r$, so

$$\frac{p_- p_+}{p_- + p_+} = r(1 - r),$$

and the bound reduces to

$$\text{Regret} \geq \frac{\mu}{2} r(1 - r)\Delta^2.$$

□

B.4 Perturbation-Aware Risk Bound

We next prove Theorem 2.2. The only regularity condition needed is local Lipschitzness of the contrastive surrogate on the region visited during training.

Assumption B.4 (Local Lipschitz surrogate)**.** For every realization of T, the loss $\ell(\cdot; T)$ is L-Lipschitz with respect to its embedding argument on the compact region $\mathcal{Z}$ visited during training:

$$|\ell(u; T) - \ell(v; T)| \leq L\|u - v\|_2, \qquad \forall u, v \in \mathcal{Z}.$$

Proof of Theorem 2.2. Fix $v, m \in [0,1]$, and $\delta \in \mathcal{Q}_v$. By Assumption B.4, for every realization of T,

$$\ell(z_v^\delta(m); T) \le \ell(z_v(m); T) + L\|z_v^\delta(m) - z_v(m)\|_2.$$

Taking expectation over T gives

$$\mathbb{E}_T[\ell(z_v^\delta(m); T)] \le \mathcal{E}_v(m) + L\|z_v^\delta(m) - z_v(m)\|_2.$$

Taking the supremum over $\delta \in \mathcal{Q}_v$ yields

$$R_v^{\mathcal{Q}}(m) = \sup_{\delta \in \mathcal{Q}_v} \mathbb{E}_T[\ell(z_v^\delta(m); T)] \le \mathcal{E}_v(m) + L \sup_{\delta \in \mathcal{Q}_v} \|z_v^\delta(m) - z_v(m)\|_2.$$

By the definition of $S_v(m)$,

$$R_v^{\mathcal{Q}}(m) \le \mathcal{E}_v(m) + LS_v(m).$$

□

B.5 Lipschitzness of InfoNCE

We briefly justify why Assumption B.4 is mild for the normalized InfoNCE loss used in contrastive learning. Consider

$$\ell_{\mathrm{NCE}}(u, v) = -\log \frac{\exp(u^\top v/\tau)}{\sum_{k \in \mathcal{N}} \exp(u^\top k/\tau)},$$

where $\tau > 0$ is the temperature, v is the positive key, and $\mathcal{N}$ contains the positive key and negative keys. Assume all keys are ℓ_2-normalized or bounded by one, so $\|k\|_2 \le 1$ for all $k \in \mathcal{N}$ and $\|v\|_2 \le 1$. Let

$$p_k = \frac{\exp(u^\top k/\tau)}{\sum_{j \in \mathcal{N}} \exp(u^\top j/\tau)}.$$

Then

$$\nabla_u \ell_{\mathrm{NCE}}(u, v) = \frac{1}{\tau}\left(\sum_{k \in \mathcal{N}} p_k k - v\right).$$

Using the triangle inequality,

$$\|\nabla_u \ell_{\mathrm{NCE}}(u, v)\|_2 \le \frac{1}{\tau}\left(\left\|\sum_{k \in \mathcal{N}} p_k k\right\|_2 + \|v\|_2\right).$$

Since $\{p_k\}$ is a probability distribution and $\|k\|_2 \le 1$,

$$\left\|\sum_{k \in \mathcal{N}} p_k k\right\|_2 \le \sum_{k \in \mathcal{N}} p_k \|k\|_2 \le 1.$$

Therefore,

$$\|\nabla_u \ell_{\mathrm{NCE}}(u, v)\|_2 \le \frac{2}{\tau}.$$

Thus, for bounded keys and fixed temperature, the InfoNCE loss is Lipschitz with respect to the query embedding with constant at most $2/\tau$. This calculation treats the fused representation as the query while the sampled positive and negative keys are fixed by T. If the same representation also appears as a key for other samples, Lipschitzness still holds on compact normalized domains, with a constant depending on the batch size and temperature. If additional projection or normalization layers are included, the same argument yields local Lipschitzness on compact regions visited during training.

B.6 Standard-Risk-Optimal Fusion Can Be Stability-Suboptimal

Theorem 2.2 shows that the perturbation-aware upper bound contains both the standard contrastive risk and the sensitivity term. The following result gives a sufficient condition under which a coefficient that is optimal for the standard contrastive risk is suboptimal for this upper bound.

Proposition B.5 (Standard-risk-optimal fusion can be stability-suboptimal)**.** *Let*

$$m_c \in \arg\min_{m\in[0,1]} \mathcal{E}_v(m).$$

If there exists $\widetilde{m} \in [0,1]$ *such that*

$$\mathcal{E}_v(\widetilde{m}) - \mathcal{E}_v(m_c) < L\big(S_v(m_c) - S_v(\widetilde{m})\big),$$

then

$$\mathcal{E}_v(\widetilde{m}) + LS_v(\widetilde{m}) < \mathcal{E}_v(m_c) + LS_v(m_c).$$

Proof. Starting from the assumed inequality,

$$\mathcal{E}_v(\widetilde{m}) - \mathcal{E}_v(m_c) < L\big(S_v(m_c) - S_v(\widetilde{m})\big).$$

Rearranging terms gives

$$\mathcal{E}_v(\widetilde{m}) + LS_v(\widetilde{m}) < \mathcal{E}_v(m_c) + LS_v(m_c).$$

Thus, although m_c minimizes the standard contrastive risk $\mathcal{E}_v(m)$, it does not minimize the perturbation-aware upper bound. □

This proposition is only a sufficient condition. It does not claim that standard-risk minimization always fails. It states that when the reduction in sensitivity is large enough to offset a small increase in standard contrastive risk, a different coefficient can be preferable for the perturbation-aware upper bound.

A simple gap construction. Let $\mathcal{E}_v(m_c) = 0$, $S_v(m_c) = M$, and suppose there exists $\widetilde{m}$ with $\mathcal{E}_v(\widetilde{m}) = \epsilon > 0$ and $S_v(\widetilde{m}) = 0$. Then m_c is standard-risk optimal, but the difference between the two perturbation-aware upper bounds is

$$\big(\mathcal{E}_v(m_c) + LS_v(m_c)\big) - \big(\mathcal{E}_v(\widetilde{m}) + LS_v(\widetilde{m})\big) = LM - \epsilon.$$

For fixed ϵ, this gap can be made arbitrarily large as M increases. This construction illustrates the role of sensitivity, without implying that such a gap must occur in every graph.

B.7 DRO Surrogate for Local Perturbation Shifts

The main text uses the following elementary observation to justify worst-case local perturbation objectives as conservative surrogates for unknown local perturbation shifts.

Proposition B.6 (Worst-case perturbation risk upper-bounds local shifts)**.** *Let* $\mathcal{P}_v$ *be any distribution over local perturbations whose support is contained in* $\mathcal{Q}_v$*. Then, for every node* v *and every fixed* $m \in [0,1]$*,*

$$\mathbb{E}_{\delta\sim\mathcal{P}_v}\mathbb{E}_T\big[\ell(z_v^\delta(m);T)\big] \le \sup_{\delta\in\mathcal{Q}_v} \mathbb{E}_T\big[\ell(z_v^\delta(m);T)\big] = R_v^{\mathcal{Q}}(m). \tag{21}$$

Proof. Let $\mathcal{P}_v$ be any distribution over local perturbations whose support is contained in $\mathcal{Q}_v$. For any $\delta \in \mathrm{supp}(\mathcal{P}_v)$, we have $\delta \in \mathcal{Q}_v$. Therefore,

$$\mathbb{E}_T[\ell(z_v^\delta(m);T)] \le \sup_{\delta'\in\mathcal{Q}_v} \mathbb{E}_T[\ell(z_v^{\delta'}(m);T)] = R_v^{\mathcal{Q}}(m).$$

Taking expectation over $\delta \sim \mathcal{P}_v$ preserves the inequality:

$$\mathbb{E}_{\delta\sim\mathcal{P}_v}\mathbb{E}_T[\ell(z_v^\delta(m);T)] \le R_v^{\mathcal{Q}}(m).$$

□

This proposition motivates idealized worst-case perturbation training as a distributionally robust upper-bound surrogate for unknown local perturbation shifts supported on $\mathcal{Q}_v$. ASPECT-S approximates this surrogate using generated perturbations. The proposition does not prove that ASPECT-S necessarily improves robustness; it only explains why a worst-case perturbation objective is a principled stability-aware training signal when robustness to local perturbations is desired.

B.8 Channel-Wise Sensitivity Tradeoff

Define the channel-wise sensitivities

$$d_{L,v} \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z^{\delta}_{L,v} - z_{L,v}\|_2, \qquad d_{H,v} \triangleq \sup_{\delta \in \mathcal{Q}_v} \|z^{\delta}_{H,v} - z_{H,v}\|_2.$$

We prove the channel-wise bound stated in Eq. (7).

Proof. For a fixed $m \in [0,1]$,

$$z^{\delta}_v(m) - z_v(m) = m(z^{\delta}_{L,v} - z_{L,v}) + (1-m)(z^{\delta}_{H,v} - z_{H,v}).$$

Since $m \in [0,1]$, the triangle inequality gives

$$\|z^{\delta}_v(m) - z_v(m)\|_2 \le m\|z^{\delta}_{L,v} - z_{L,v}\|_2 + (1-m)\|z^{\delta}_{H,v} - z_{H,v}\|_2.$$

Taking the supremum over $\delta \in \mathcal{Q}_v$,

$$S_v(m) = \sup_{\delta \in \mathcal{Q}_v} \|z^{\delta}_v(m) - z_v(m)\|_2 \le m d_{L,v} + (1-m) d_{H,v}.$$

Combining this inequality with Theorem 2.2 yields

$$R^{\mathcal{Q}}_v(m) \le \mathcal{E}_v(m) + L S_v(m) \le \mathcal{E}_v(m) + L\big(m d_{L,v} + (1-m) d_{H,v}\big),$$

which proves Eq. (7). □

This bound makes the node-wise tradeoff explicit: the preferred spectral mixture can depend on both the standard contrastive risk and the relative local sensitivity of the low- and high-frequency channels. It does not require assuming that either channel is universally more reliable.

Remark on the fixed-gate convention. The channel-wise sensitivity decomposition assumes that the fusion coefficient is fixed when perturbations are generated. This matches ASPECT-S, where the gate is computed on the clean graph and held fixed during the inner perturbation search. If the gate were recomputed on the perturbed graph, an additional gate-variation term would appear:

$$z^{\delta}_v(m^{\delta}) - z_v(m) = m(z^{\delta}_{L,v} - z_{L,v}) + (1-m)(z^{\delta}_{H,v} - z_{H,v}) + (m^{\delta} - m)(z^{\delta}_{L,v} - z^{\delta}_{H,v}).$$

Thus, the fixed-gate analysis corresponds to the perturbation-generation procedure used in ASPECT-S.

B.9 Entropy-Regularized Interpretation of the Policy Target

This subsection gives a simple interpretation of the policy target used in the utility-aware spectral policy regularizer. The main text shows that, when robustness to local perturbations is desired, the preferred channel mixture may depend on both channel-wise utility and channel-wise sensitivity. In practice, these quantities are not known and must be estimated from training signals. The policy regularizer used by ASPECT can be viewed as a plug-in instantiation to an entropy-regularized channel-cost minimization problem.

For a node v, let $B_{L,v}$ and $B_{H,v}$ denote estimated costs for the low- and high-frequency channels, respectively. These costs may represent only utility evidence, as in ASPECT, or utility plus empirical sensitivity evidence, as in ASPECT-S. Consider the entropy-regularized channel-selection problem

$$\min_{q_v \in \Delta_2} \sum_{c \in \{L,H\}} q_{c,v} B_{c,v} + \tau_g \sum_{c \in \{L,H\}} q_{c,v} \log q_{c,v}, \tag{22}$$

where $\Delta_2 = \{q_v : q_{L,v} \ge 0, q_{H,v} \ge 0, q_{L,v} + q_{H,v} = 1\}$ and $\tau_g > 0$ is a temperature parameter. The first term prefers channels with smaller estimated costs, while the negative-entropy regularization controls the sharpness of the target policy through τ_g.

Proposition B.7 (Entropy-regularized plug-in channel policy)**.** *For any $B_{L,v}, B_{H,v} \in \mathbb{R}$ and $\tau_g > 0$, the optimization problem in Eq.* (22) *has a unique minimizer given by*

$$q^{\star}_{c,v} = \frac{\exp(-B_{c,v}/\tau_g)}{\exp(-B_{L,v}/\tau_g) + \exp(-B_{H,v}/\tau_g)}, \qquad c \in \{L,H\}. \tag{23}$$

In particular, the optimal low-frequency weight is

$$q^{\star}_{L,v} = \frac{\exp(-B_{L,v}/\tau_g)}{\exp(-B_{L,v}/\tau_g) + \exp(-B_{H,v}/\tau_g)}. \tag{24}$$

Proof. The objective in Eq. (22) is strictly convex on the probability simplex because $\tau_g > 0$ and the negative entropy term is strictly convex in the interior of the simplex. Introducing a Lagrange multiplier λ for the constraint $\sum_c q_{c,v} = 1$, the first-order condition for each $c \in \{L, H\}$ is

$$B_{c,v} + \tau_g(1 + \log q_{c,v}) + \lambda = 0.$$

Rearranging gives

$$q_{c,v} = C \exp(-B_{c,v}/\tau_g),$$

where $C = \exp(-(1 + \lambda/\tau_g))$ is a normalization constant independent of c. Enforcing $\sum_c q_{c,v} = 1$ gives Eq. (23). Strict convexity implies uniqueness. □

Proposition B.7 shows that a softmax over negative channel costs is the exact solution of an entropy-regularized plug-in policy problem. ASPECT instantiates the channel costs using channel-wise standard contrastive evidence:

$$B_{c,v} = b^{\mathrm{core}}_{c,v} \triangleq \mathrm{Norm}_\ell\big(\ell^{\mathrm{std}}_{c,v}\big), \qquad c \in \{L, H\}, \tag{25}$$

where $\ell^{\mathrm{std}}_{c,v}$ is the standard contrastive loss associated with channel c at node v, and $\mathrm{Norm}_\ell(\cdot)$ denotes the normalization used to make channel-wise losses comparable. This gives the utility-aware target

$$\widetilde{m}_v = \frac{\exp(-b^{\mathrm{core}}_{L,v}/\tau_g)}{\exp(-b^{\mathrm{core}}_{L,v}/\tau_g) + \exp(-b^{\mathrm{core}}_{H,v}/\tau_g)}. \tag{26}$$

Thus, in ASPECT, the auxiliary target assigns larger weight to the channel with smaller estimated contrastive cost, while remaining soft due to entropy regularization.

In ASPECT-S, when generated perturbations are available, the same plug-in form is augmented with empirical channel-wise sensitivity estimates:

$$B_{c,v} = b^{\mathrm{stable}}_{c,v} \triangleq \mathrm{Norm}_\ell\big(\ell^{\mathrm{std}}_{c,v}\big) + \lambda_s \,\mathrm{Norm}_d\big(\widehat{d}_{c,v}\big), \qquad c \in \{L, H\}, \tag{27}$$

where $\widehat{d}_{c,v}$ denotes the empirical sensitivity estimate of channel c at node v, $\mathrm{Norm}_d(\cdot)$ normalizes sensitivity scores, and $\lambda_s \geq 0$ controls the contribution of the sensitivity term. The corresponding utility–sensitivity-aware target is

$$\widetilde{m}_v = \frac{\exp(-b^{\mathrm{stable}}_{L,v}/\tau_g)}{\exp(-b^{\mathrm{stable}}_{L,v}/\tau_g) + \exp(-b^{\mathrm{stable}}_{H,v}/\tau_g)}. \tag{28}$$

Finally, the learned node-wise spectral policy m_v is encouraged to match the plug-in target through an auxiliary binary cross-entropy regularizer. In implementation, both $\widetilde{m}_v$ and the channel costs used to construct it are detached from the computational graph. Thus, this loss updates the predicted gate through m_v rather than through the pseudo-target. Specifically, we use

$$\mathcal{L}_{\mathrm{pol}} = -\frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \left[\mathrm{sg}(\widetilde{m}_v) \log m_v + \big(1 - \mathrm{sg}(\widetilde{m}_v)\big) \log(1 - m_v) \right], \tag{29}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. Equivalently, $\mathcal{L}_{\mathrm{pol}}$ is the cross-entropy between a detached Bernoulli pseudo-target with parameter $\widetilde{m}_v$ and the predicted Bernoulli gate with parameter m_v. For a fixed target $\widetilde{m}_v$, this loss is minimized when $m_v = \widetilde{m}_v$. Therefore, the policy regularizer encourages the learned gate to follow the entropy-regularized plug-in channel policy, while preventing gradients from changing the channel-cost estimates through the pseudo-target.

B.10 Treatment of T and View Sampling

In the main text, $z_v(m)$ and $z^\delta_v(m)$ are treated as representations conditioned on the clean and perturbed graphs, respectively, while T captures randomness in the contrastive objective, such as positive-view and negative-sample selection. This convention separates representation perturbations from sampling randomness in the loss.

If view sampling also changes the representation itself, the same proof strategy can be applied by conditioning on T and treating $z_v(m; T)$ and $z^\delta_v(m; T)$ as the sampled-view representations. In that case, one can define sensitivity either conditionally for each T or uniformly over the sampled views. The resulting bounds have the same form, with $S_v(m)$ replaced by the corresponding conditional or uniform sensitivity term.

C Additional Experiments

This section provides a pretraining-seed sensitivity diagnostic, arXiv-year and hyperparameter-sensitivity experiments, the full clean-performance comparison, the full fixed-budget perturbation comparison, and variable-budget perturbation curves. These results complement the selected-baseline tables in Section 4 and support the same conclusions.

C.1 Large-Scale Evaluation on arXiv-year

We evaluated arXiv-year using the ogbn-arxiv graph and features with non-homophilous year-derived labels. The graph has 169,343 nodes, 128-dimensional features, and 2,315,598 nonzeros in the symmetrized adjacency. All methods used the same data splits, an NVIDIA RTX PRO 6000 Blackwell GPU (96 GiB), hidden dimension 256, and Chebyshev order $K = 10$. Table 4 reports mean test accuracy and standard deviation across the splits. Epoch times exclude evaluation and data transfer and were measured after timing warm-up; peak GPU memory was measured consistently across methods.

Table 4: arXiv-year test accuracy, epoch time, and peak GPU memory.

Method	Test accuracy (%)	Time / epoch	Peak GPU memory
PolyGCL	43.09 ± 0.29	0.5012 s	50.79 GiB
Global Fusion	41.80 ± 0.21	0.3374 s	9.67 GiB
ASPECT	43.65 ± 0.32	0.3428 s	9.76 GiB
ASPECT-S	42.78 ± 0.30	0.5039 s	14.02 GiB

ASPECT exceeds PolyGCL by 0.56 points and the matched Global Fusion ablation by 1.85 points. PolyGCL and ASPECT use different contrastive-scoring implementations, so their raw runtime and memory differences do not isolate the overhead of node-wise fusion. In the matched ASPECT–Global Fusion comparison, both methods use the same dual-channel encoder and sampled-negative contrastive objective; the node-wise gate adds 1.6% runtime and 0.09 GiB peak memory. ASPECT-S alternates regular and perturbation epochs: a regular epoch takes 0.3424 s, and a perturbation epoch, including inner generation and the outer update, takes 1.1497 s. Perturbations are regenerated once every five epochs, yielding 0.5039 s amortized per epoch. The 14.02 GiB peak is measured during a perturbation epoch; regular epochs have essentially the same memory profile as ASPECT.

C.2 Hyperparameter Sensitivity

We conducted a one-at-a-time sensitivity analysis on Cora and Squirrel, representing homophilic and heterophilic graphs. Only the parameter under study was varied; all other hyperparameters stayed at their reported values. We used the same data splits, random seeds, and, where applicable, evaluation perturbations across settings. For dataset D, a star denotes the original Optuna-selected value. We evaluated

$$\frac{\tau_{g,D}}{\tau^*_{g,D}} \in \{0.25, 0.5, 1, 2, 4\}, \qquad \frac{\lambda_{s,D}}{\lambda^*_{s,D}} \in \{0, 0.5, 1, 2, 4\}, \qquad \frac{\lambda_{\mathrm{ray},D}}{\lambda^*_{\mathrm{ray},D}} \in \{0, 0.5, 1, 2, 4\}. \tag{30}$$

The $1\times$ entries reproduce the original Optuna-selected results from the main clean and perturbation tables. Every other setting was independently run using the same code, schedule, splits, seeds, evaluation perturbations, and pipeline. Setting $\lambda_s = 0$ removes sensitivity evidence; $\lambda_{\mathrm{ray}} = 0$ removes only the Rayleigh bias. We report clean accuracy for τ_g, the gate parameter in ASPECT, and perturbed accuracy for λ_s and λ_{ray}, which control the stability-aware mechanisms in ASPECT-S.

Accuracy varies by less than one point across the τ_g sweep on both datasets. Removing sensitivity evidence ($\lambda_s = 0$) reduces perturbed accuracy by 2.49/3.18 points on Cora/Squirrel. Removing the Rayleigh bias ($\lambda_{\mathrm{ray}} = 0$) reduces it by 1.62/2.02 points. The original $1\times$ configurations are best or comparable across all sweeps.

Table 5: Gate-temperature sensitivity: clean node-classification accuracy (mean ± standard deviation, %). Each multiplier applies to the dataset-specific Optuna-selected τ_g^*.

Dataset	$0.25\times$	$0.5\times$	$1\times$	$2\times$	$4\times$
Cora	88.07 ± 1.26	88.64 ± 1.08	88.94 ± 1.10	88.66 ± 1.38	88.14 ± 1.37
Squirrel	58.41 ± 1.57	58.84 ± 1.63	59.39 ± 1.76	59.04 ± 1.82	58.62 ± 1.62

Table 6: ASPECT-S sensitivity: perturbed node-classification accuracy (mean ± standard deviation, %). Each multiplier applies to the dataset-specific Optuna-selected parameter value.

Parameter	Dataset	$0\times$	$0.5\times$	$1\times$	$2\times$	$4\times$
λ_s	Cora	82.81 ± 1.39	84.82 ± 1.02	85.30 ± 0.72	85.42 ± 0.83	83.50 ± 1.08
λ_s	Squirrel	46.99 ± 1.91	48.16 ± 1.31	50.17 ± 0.92	48.29 ± 1.18	47.06 ± 0.87
λ_{ray}	Cora	83.68 ± 1.19	84.84 ± 1.01	85.30 ± 0.72	85.01 ± 0.73	84.72 ± 0.60
λ_{ray}	Squirrel	48.15 ± 1.94	48.98 ± 1.64	50.17 ± 0.92	50.06 ± 1.33	48.75 ± 0.70

C.3 Pretraining-Seed Sensitivity of the Spectral Policy

We assess how reliably the node-wise policy learns across ten pretraining seeds. For each dataset, the linear-evaluation data split and evaluation seed are fixed across runs; only the pretraining seed changes. We compare ASPECT with the same node-wise architecture trained without the policy regularizer. Table 7 reports the standard deviation of the final linear-evaluation accuracy across the ten runs. The smaller spread for ASPECT on both Cora and Squirrel shows that channel-wise contrastive evidence makes the learned policy less sensitive to pretraining randomness. The main performance tables instead aggregate across ten data splits.

Table 7: Standard deviation of linear-evaluation accuracy (percentage points) across ten pretraining seeds, with the evaluation split and seed fixed.

Variant	**Cora**	**Squirrel**
ASPECT	0.36	0.52
w/o PolicyReg	2.51	2.79

C.4 Full Clean Performance Comparison

Table 8 reports the complete clean linear-evaluation results over all baselines considered in our experiments. The selected-baseline table in the main text is a compact summary of this full comparison. Across the full set of baselines, ASPECT achieves the best performance on 8 out of 9 datasets, while ASPECT-S remains competitive under the clean evaluation protocol.

C.5 Full Perturbation Performance Comparison

Table 9 gives the complete fixed-budget perturbation results. In this setting, graph-structure and feature perturbations are applied jointly: Metattack perturbs 10% of edges, and feature masking corrupts 10% of node features. ASPECT-S obtains the best perturbed accuracy on all evaluated datasets and the lowest average relative accuracy drop, consistent with the main-text summary.

Figure 3 further evaluates the variable-budget setting. We jointly increase the edge perturbation ratio and feature masking ratio from 0% to 25%. ASPECT-S consistently maintains higher accuracy as the perturbation budget increases, indicating that ASPECT-S improves degradation behavior beyond the fixed-budget setting.

Table 8: Full clean node classification accuracy (mean ± standard deviation, %) under the linear evaluation protocol. This table includes all baselines considered in our experiments. Boldface indicates the best performance and underline indicates the second-best performance.

Methods	Homophilic Datasets			Heterophilic Datasets					
	Cora	Citeseer	Pubmed	Cornell	Texas	Wisconsin	Actor	Chameleon	Squirrel
DGI	85.88 ± 0.95	76.44 ± 0.84	82.13 ± 0.24	70.82 ± 2.71	81.48 ± 2.79	75.00 ± 4.22	32.09 ± 1.18	58.23 ± 0.70	38.80 ± 0.76
MVGRL	87.36 ± 0.64	78.70 ± 0.64	86.30 ± 0.23	67.70 ± 4.45	73.11 ± 4.47	74.25 ± 2.43	32.98 ± 0.53	57.75 ± 1.20	40.25 ± 1.14
GMI	85.09 ± 1.13	76.38 ± 0.70	83.06 ± 0.34	62.79 ± 3.85	68.03 ± 2.02	62.13 ± 2.88	32.37 ± 1.16	62.47 ± 1.52	39.82 ± 0.94
GGD	87.21 ± 1.18	79.25 ± 1.06	85.38 ± 0.25	80.33 ± 1.80	82.62 ± 1.41	73.25 ± 3.28	32.27 ± 1.17	57.64 ± 1.65	40.87 ± 0.93
GraphCL	86.54 ± 1.34	78.99 ± 1.95	85.16 ± 0.60	61.48 ± 4.69	66.07 ± 3.42	60.63 ± 2.19	32.45 ± 1.13	58.49 ± 1.23	42.92 ± 0.96
GRACE	83.27 ± 0.74	73.79 ± 0.57	81.71 ± 0.14	60.66 ± 2.94	75.74 ± 3.12	72.13 ± 1.99	31.97 ± 1.13	59.52 ± 2.65	42.68 ± 1.10
GCA	84.09 ± 0.85	75.23 ± 1.19	82.01 ± 0.34	53.11 ± 4.01	81.97 ± 1.58	73.50 ± 2.85	31.13 ± 1.11	65.54 ± 1.10	47.13 ± 0.93
GREET	85.16 ± 0.77	79.06 ± 1.34	85.64 ± 0.28	78.36 ± 2.77	78.03 ± 3.94	84.63 ± 2.10	37.12 ± 0.67	60.57 ± 1.03	42.80 ± 1.01
BGRL	84.45 ± 0.66	74.84 ± 1.44	83.06 ± 0.29	59.84 ± 3.12	69.84 ± 2.91	62.88 ± 3.52	32.48 ± 1.16	64.09 ± 3.44	47.02 ± 0.95
GBT	84.89 ± 1.11	76.59 ± 0.81	86.10 ± 0.29	59.18 ± 3.54	72.79 ± 2.79	62.38 ± 2.71	34.34 ± 1.10	68.77 ± 1.23	48.86 ± 0.87
CCA-SSG	87.39 ± 0.89	79.60 ± 1.01	84.95 ± 0.26	78.69 ± 4.61	87.87 ± 1.89	82.88 ± 3.58	34.86 ± 1.13	59.84 ± 1.21	41.50 ± 1.12
SP-GCL	82.99 ± 1.18	75.54 ± 1.06	85.74 ± 0.21	69.41 ± 1.49	69.76 ± 1.23	69.34 ± 0.77	35.92 ± 0.67	69.23 ± 1.23	53.05 ± 1.05
HLCL	85.53 ± 1.03	76.79 ± 0.60	85.13 ± 0.18	64.00 ± 8.98	78.38 ± 5.08	79.50 ± 4.50	40.56 ± 0.70	63.86 ± 1.34	44.49 ± 0.68
PolyGCL	87.57 ± 0.62	79.81 ± 0.85	87.15 ± 0.27	82.62 ± 3.11	88.03 ± 1.80	85.50 ± 1.88	41.15 ± 0.88	71.62 ± 0.96	56.49 ± 0.72
S3GCL	87.04 ± 1.25	77.48 ± 0.80	86.03 ± 0.37	81.27 ± 3.67	86.12 ± 3.91	84.56 ± 2.71	40.06 ± 1.58	71.88 ± 1.91	56.90 ± 1.37
RDGI	83.53 ± 1.23	78.99 ± 0.80	80.89 ± 1.55	67.21 ± 6.06	69.01 ± 4.59	56.75 ± 4.12	32.74 ± 1.27	59.95 ± 1.11	42.71 ± 0.70
ARIEL	87.30 ± 0.71	79.53 ± 0.61	86.42 ± 0.47	70.70 ± 2.46	76.19 ± 5.02	71.15 ± 2.38	37.68 ± 1.03	64.53 ± 1.47	42.42 ± 1.53
ASPECT	**88.94** ± 1.10	**81.82** ± 0.90	**87.75** ± 1.03	**89.18** ± 2.77	**91.76** ± 2.09	**90.03** ± 2.09	**42.64** ± 1.55	**72.88** ± 1.90	59.39 ± 1.76
ASPECT-S	88.69 ± 0.72	81.30 ± 0.85	86.92 ± 0.69	88.68 ± 2.26	90.41 ± 1.97	88.00 ± 2.12	41.73 ± 1.30	72.48 ± 1.43	**59.91** ± 0.90

Table 9: Full node classification accuracy (mean ± standard deviation, %) under fixed-budget joint graph-structure and feature perturbations. Metattack perturbs 10% of edges and feature masking corrupts 10% of node features. This table includes all baselines considered in the perturbation evaluation. Boldface indicates the best result and underline indicates the second-best result.

Methods	Homophilic Datasets			Heterophilic Datasets			Avg. Drop (%)
	Cora	Citeseer	Pubmed	Actor	Chameleon	Squirrel	
DGI	79.62 ± 0.62	72.25 ± 0.85	74.29 ± 1.01	30.28 ± 1.32	51.47 ± 0.70	32.94 ± 0.73	9.11
MVGRL	77.93 ± 0.76	70.31 ± 1.00	73.57 ± 0.49	27.00 ± 0.52	54.62 ± 1.09	39.31 ± 1.13	10.35
GMI	79.23 ± 0.56	70.67 ± 0.85	73.51 ± 0.66	28.88 ± 0.96	52.01 ± 1.27	32.07 ± 1.15	12.14
GGD	80.72 ± 0.61	71.00 ± 0.83	72.97 ± 0.70	30.29 ± 1.60	50.92 ± 1.51	32.23 ± 1.19	11.89
GraphCL	78.54 ± 0.89	72.40 ± 1.19	73.94 ± 0.70	31.04 ± 0.56	49.93 ± 0.88	31.69 ± 1.44	12.65
GRACE	77.08 ± 1.28	70.67 ± 0.86	75.25 ± 0.60	30.78 ± 0.71	51.38 ± 1.75	32.76 ± 1.07	10.03
GCA	76.39 ± 0.92	56.55 ± 1.31	71.32 ± 0.87	31.87 ± 0.97	58.75 ± 1.09	37.20 ± 0.90	12.68
GREET	78.80 ± 1.45	75.44 ± 0.59	79.47 ± 0.57	34.46 ± 1.23	51.77 ± 1.55	35.64 ± 1.32	9.61
BGRL	75.04 ± 0.81	68.10 ± 0.83	73.29 ± 1.03	30.19 ± 1.23	53.00 ± 1.20	35.05 ± 1.09	13.62
GBT	79.84 ± 0.46	72.07 ± 0.89	75.60 ± 1.30	33.10 ± 1.23	57.59 ± 1.41	38.93 ± 0.51	10.71
CCA-SSG	82.79 ± 1.28	74.88 ± 0.72	77.01 ± 0.90	30.70 ± 0.77	49.63 ± 1.09	31.23 ± 1.44	12.57
SP-GCL	76.32 ± 1.11	70.12 ± 1.07	74.76 ± 0.79	30.77 ± 0.76	62.02 ± 1.72	41.94 ± 1.32	12.29
PolyGCL	83.18 ± 0.78	72.51 ± 1.25	77.82 ± 0.83	37.35 ± 0.90	59.01 ± 1.35	40.89 ± 1.40	13.22
S3GCL	80.31 ± 0.62	71.72 ± 1.40	79.46 ± 1.57	36.03 ± 1.28	59.89 ± 1.99	40.29 ± 1.75	13.12
RDGI	78.85 ± 0.96	73.92 ± 0.68	74.12 ± 1.41	30.37 ± 1.47	52.66 ± 0.94	34.00 ± 0.63	10.03
ARIEL	84.80 ± 1.01	76.17 ± 1.39	81.08 ± 0.95	32.33 ± 0.43	56.18 ± 1.08	36.09 ± 1.11	9.22
ASPECT	84.08 ± 1.41	77.44 ± 0.86	82.38 ± 0.52	36.14 ± 0.47	64.18 ± 1.93	46.37 ± 1.08	11.01
ASPECT-S	**85.30** ± 0.72	**78.86** ± 0.70	**84.93** ± 0.44	**39.20** ± 0.63	**66.97** ± 1.70	**50.17** ± 0.92	**6.51**

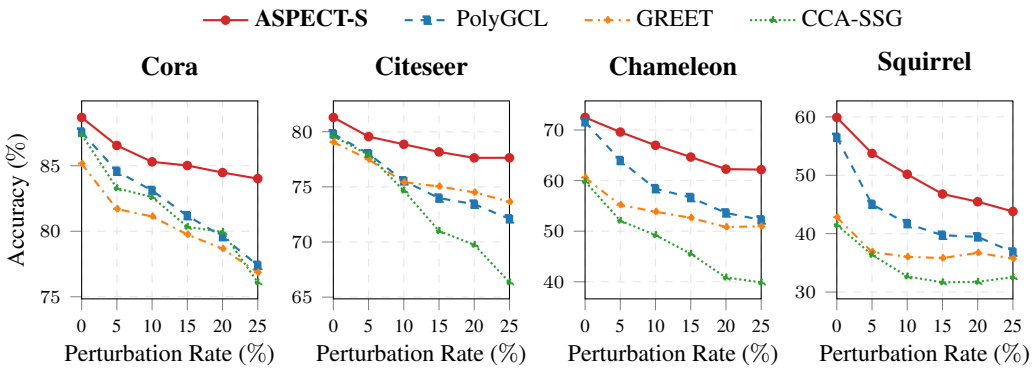

Figure 3: **Variable-budget perturbation performance.** Classification accuracy (%) as the perturbation budget increases. For each budget, graph-structure perturbations are generated by Metattack and feature perturbations are applied by masking the same percentage of node features. ASPECT-S degrades more gracefully than representative baselines across both homophilic and heterophilic datasets, suggesting that ASPECT-S improves performance under the evaluated joint graph-structure and feature perturbation protocol.

D Dataset Details

This paper relies on several publicly available datasets. We provide detailed information to facilitate their usage and verification.

D.1 Dataset Descriptions and Sources

We conduct our experiments on the following widely used benchmark datasets, all drawn from existing literature and publicly available for research purposes:

- **Homophilic Datasets:** `Cora`, `Citeseer`, and `Pubmed` (Sen et al., 2008). These are standard citation networks commonly used for evaluating graph learning models. In these graphs, nodes represent papers and edges represent citation relationships between two papers. The features consist of bag-of-words representations of the papers, while the labels indicate the research topic of each paper.
- **Heterophilic Datasets:** `Chameleon`, `Squirrel` (Rozemberczki et al., 2021) are two heterophilic networks based on Wikipedia. The nodes denote web pages in Wikipedia and edges denote links between them. The features consist of informative nouns in the Wikipedia pages, and labels indicate the average traffic of the web pages. `Actor` (Pei et al., 2020) is an actor co-occurrence network where nodes represent actors and edges indicate co-occurrence on the same Wikipedia page. Node features are keyword-based representations extracted from Wikipedia pages, and labels correspond to five actor-related categories derived from the page content. It is commonly used as a heterophilic graph benchmark. `Cornell`, `Texas`, and `Wisconsin` (Pei et al., 2020) are three heterophilic networks originating from the WebKB project, where nodes are web pages of the computer science departments of different universities and edges are hyperlinks between them. The features of each page are represented as bag-of-words, and the labels indicate the types of web pages.

All datasets were sourced from their official or commonly accepted repositories (e.g., PyTorch Geometric, Deep Graph Library). No custom or novel datasets were created or used for this work. The motivation for selecting these datasets is to cover a broad spectrum of graph properties, including both homophilic and heterophilic structures, which is crucial for evaluating spectral graph contrastive learning methods with node-adaptive fusion and ASPECT-S.

D.2 Data Preprocessing and Partitioning

For all datasets, raw node features are used. We symmetrize adjacency matrices, treat them as unweighted, and add self-loops before model training. We strictly adhere to the standard experimental

protocol of 10 random 60%/20%/20% train/validation/test splits for node classification, as proposed by Chien et al. (2020) and commonly used in graph representation learning literature. The random seeds for these splits are fixed and consistent across all runs and baselines to ensure a fair and reproducible comparison of results. No additional dataset preprocessing beyond these steps was applied. Training-time graph augmentations used by contrastive objectives are described in Section 3 and follow the configurations in Appendix E.

E Implementation Details

For reproducibility, we provide implementation details for our experiments.

E.1 Baselines

We compare ASPECT against representative self-supervised GCL methods from four families.

(i) General augmentation-based GCL: DGI (Velickovic et al., 2019), MVGRL (Hassani & Khasahmadi, 2020), GMI (Peng et al., 2020), GGD (Zheng et al., 2022b), GraphCL (You et al., 2020), GRACE (Zhu et al., 2020b), GCA (Zhu et al., 2021), and GREET (Liu et al., 2023).

(ii) Invariance-keeping / predictor-based GCL: BGRL (Thakoor et al., 2021), GBT (Bielak et al., 2022), and CCA-SSG (Zhang et al., 2021).

(iii) Heterophily- and spectral-oriented GCL: SP-GCL (Wang et al., 2023), HLCL (Yang & Mirzasoleiman, 2024), PolyGCL (Chen et al., 2024), and S3GCL (Wan et al., 2024). Among them, **PolyGCL** is the most direct external control for our theory: it adopts dual spectral channels but relies on *node-agnostic* fusion.

(iv) Robust / adversarial representation learning on graphs: RDGI (Xu et al., 2022) and ARIEL (Feng et al., 2024).

E.2 Reproducibility Note.

We primarily utilize official open-source implementations for all baselines (see Table 10 for URLs). Regarding **HLCL** (Yang & Mirzasoleiman, 2024), as we could not obtain an official implementation compatible with our evaluation pipeline at the time of our experiments, we report its clean performance directly from the PolyGCL paper (Chen et al., 2024), which follows the same reported evaluation protocol. Consequently, HLCL is excluded from the perturbation evaluation as we could not subject it to our specific Metattack pipeline. Similarly, recent global fusion methods such as **DPGCL** (Huang et al., 2024) and **LOHA** (Zou et al., 2025) are excluded from comparison, since we could not obtain an official implementation compatible with our protocol at the time of our experiments.

In our perturbation evaluation protocol, for each split, Metattack is run using only the training labels of the corresponding split and a fixed surrogate model. Perturbed graphs are generated independently for each dataset/split. Feature perturbation masks feature entries at the specified ratio. No perturbed validation data is used for hyperparameter selection.

E.3 Spectral Filter Parameterization

This subsection describes the spectral filter parameterization used to construct the low- and high-frequency views in ASPECT. The parameterization follows the monotone spectral response design used in PolyGCL (Chen et al., 2024). It is an implementation choice for producing complementary low- and high-frequency channels, while the main contribution of ASPECT lies in how these channels are adaptively fused at the node level.

We approximate each spectral filter by a truncated Chebyshev polynomial of order K. Instead of directly learning the polynomial coefficients, we first learn the filter values at Chebyshev nodes and then recover the polynomial coefficients analytically. Let $\{x_j\}_{j=0}^{K}$ denote the Chebyshev nodes and let $\gamma_j = g(x_j)$ be the filter response at node x_j. To encourage a monotonically non-decreasing high-pass response and a monotonically non-increasing low-pass response, we parameterize the filter

values using learnable increments $\{\delta_j^H\}_{j=0}^K$ and $\{\delta_j^L\}_{j=0}^K$:

$$\begin{aligned} \gamma_i^H &= \sum_{j=0}^{i} \mathrm{ReLU}(\delta_j^H), \\ \gamma_i^L &= \mathrm{ReLU}(\delta_0^L) - \sum_{j=1}^{i} \mathrm{ReLU}(\delta_j^L), \end{aligned} \qquad i = 0, \dots, K. \tag{31}$$

This prefix-sum parameterization constrains the high-pass response to increase with the spectral coordinate and the low-pass response to decrease with the spectral coordinate.

Given the filter responses $\{\gamma_j\}_{j=0}^K$, the Chebyshev coefficients are recovered by the discrete Chebyshev transform:

$$w_k = \begin{cases} \dfrac{1}{K+1} \displaystyle\sum_{j=0}^{K} \gamma_j T_0(x_j), & k = 0, \\ \dfrac{2}{K+1} \displaystyle\sum_{j=0}^{K} \gamma_j T_k(x_j), & k = 1, \dots, K. \end{cases} \tag{32}$$

where $T_k(\cdot)$ is the k-th Chebyshev basis. In implementation, we apply this recovery separately to the low- and high-frequency channels, yielding coefficients $\{w_k^L\}_{k=0}^K$ and $\{w_k^H\}_{k=0}^K$.

Let $\mathbf{L}$ be the normalized graph Laplacian and $\widetilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}$ be the rescaled Laplacian. The two spectral views are then computed as

$$\begin{aligned} \mathbf{Z}_L &= f_\theta\left(\sum_{k=0}^{K} w_k^L T_k(\widetilde{\mathbf{L}})\mathbf{X}\right), \\ \mathbf{Z}_H &= f_\theta\left(\sum_{k=0}^{K} w_k^H T_k(\widetilde{\mathbf{L}})\mathbf{X}\right), \end{aligned} \tag{33}$$

where $f_\theta(\cdot)$ is a shared projection network. The Chebyshev recursion allows Eq. (33) to be computed using sparse matrix-vector multiplications without eigendecomposition. The resulting $\mathbf{Z}_L$ and $\mathbf{Z}_H$ provide complementary low- and high-frequency views, which are subsequently combined by the node-wise spectral policy in ASPECT.

E.4 ASPECT-S Perturbation Generation

Rayleigh-based spectral search bias. ASPECT-S uses a Rayleigh-based term to guide the inner perturbation-generation step toward perturbations that induce informative spectral-profile shifts. For a perturbed graph $G_{\mathrm{pert}} = (A_{\mathrm{pert}}, X_{\mathrm{pert}})$, we define

$$\Phi_{\mathrm{ray}} = \mathcal{R}(A_{\mathrm{pert}}, Z_L^{\mathrm{pert}}) - \mathcal{R}(A_{\mathrm{pert}}, Z_H^{\mathrm{pert}}), \qquad \mathcal{R}(A, Z) = \frac{\mathrm{Tr}(Z^\top L_A Z)}{\mathrm{Tr}(Z^\top Z)},$$

where L_A is the normalized graph Laplacian associated with A. The Rayleigh energy measures the Dirichlet smoothness of a representation on the current perturbed graph: lower values correspond to smoother graph signals, whereas larger values indicate stronger high-frequency variation. During perturbation generation, ASPECT-S maximizes

$$J_{\mathrm{pert}} = L_{\mathrm{gen}} + \lambda_{\mathrm{ray}} \Phi_{\mathrm{ray}}$$

over the structural and feature perturbation variables, while keeping the encoder parameters and the clean-graph gate fixed. Since the nominal low-frequency channel is expected to have lower Rayleigh energy than the high-frequency channel, maximizing Φ_{ray} encourages perturbations that disturb this relative spectral profile, for example by increasing the Dirichlet energy of the perturbed low-frequency representation relative to the perturbed high-frequency representation. This makes the generated perturbations more informative for exposing channel-dependent sensitivity. Importantly, Φ_{ray} is not used as a robustness certificate and does not assume that either spectral channel is inherently fragile. It is a spectral search bias for constructing useful perturbation probes; the node-wise policy is still trained from the empirical utility and sensitivity evidence in the policy target. In implementation, the Rayleigh term is computed with sparse Laplacian-vector products and does not require eigendecomposition.

Table 10: Codes & commit numbers.

Method	URL	Commit
DGI	https://github.com/PetarV-/DGI	61baf67
MVGRL	https://github.com/kavehhassani/mvgrl	628ed2b
GMI	https://github.com/zpeng27/GMI	3491e8c
GGD	https://github.com/zyzisastudyreallyhardguy/graph-group-discrimination	7cf72db
GRACE	https://github.com/CRIPAC-DIG/GRACE	51b4496
GCA	https://github.com/CRIPAC-DIG/GCA	cd6a9f0
GraphCL	https://github.com/Shen-Lab/GraphCL	a0c8c97
GREET	https://github.com/yixinliu233/GREET	8bcc940
BGRL	https://github.com/nerdslab/bgrl	60f9f19
GBT	https://github.com/pbielak/graph-barlow-twins	ec62580
CCA-SSG	https://github.com/hengruizhang98/CCA-SSG	cea6e73
SP-GCL	https://github.com/haonan3/SPGCL	58caefa
POLYGCL	https://github.com/ChenJY-Count/PolyGCL	ec246bc
S3GCL	https://github.com/GuanchengWan/S3GCL	35c4cfc
RDGI	https://github.com/galina0217/robustgraph	2ee6abb
ARIEL	https://github.com/Shengyu-Feng/ARIEL	e761cb8

E.5 ASPECT-S Optimization and Implementation Details

This subsection describes the engineering optimizations used to reduce computational and memory overhead of ASPECT-S.

Sparse perturbation candidates. For structural perturbations, ASPECT-S does not search over all possible node pairs. Instead, we construct a sparse candidate edge set $\mathcal{E}_{\text{cand}}$. Candidate edges may include existing edges, sampled local non-edges, or edges around nodes used in the stability-aware loss. Perturbation variables are defined and updated only on $\mathcal{E}_{\text{cand}}$. Feature perturbations are constrained by a prescribed feature budget, such as a mask ratio or a sampled set of candidate feature dimensions. This reduces structural perturbation search from dense $O(|V|^2)$ candidates to a sparse set, lowers memory and runtime, and keeps perturbations local and scalable.

ASPECT warmup and perturbation interval. We first train ASPECT for a warmup period without ASPECT-S perturbation generation. After warmup, perturbations are generated only every r epochs. On non-perturbation epochs, we optimize the ASPECT objective; on perturbation epochs, we generate perturbations and optimize the ASPECT-S objective:

$$\text{if } t \le T_{\text{warm}}: \quad \text{train ASPECT only,}$$

$$\text{else if } (t - T_{\text{warm}}) \bmod r = 0: \quad \text{generate perturbations and train ASPECT-S,}$$

$$\text{else}: \quad \text{train ASPECT only.}$$

This schedule reduces expensive perturbation-generation calls, lets the node-wise spectral policy learn first under the ASPECT objective, and prevents ASPECT-S from dominating training cost.

Detached policy target and no-gradient channel scores. The policy regularizer uses a soft target $\widetilde{m}_v$ computed from channel-wise costs. When constructing this target, we detach channel embeddings, channel-wise standard losses, and empirical sensitivity estimates. Thus, only the predicted gate m_v receives gradients from $\mathcal{L}_{\text{pol}}$; the target $\widetilde{m}_v$, channel costs $b_{c,v}$, standard channel losses $\ell^{\text{std}}_{c,v}$, and empirical sensitivity estimates $\widehat{d}_{c,v}$ are treated as no-gradient quantities. A code-style sketch is shown below:

```
with torch.no_grad():
    loss_L = channel_score(z_L.detach(), z_L_aug.detach())
    loss_H = channel_score(z_H.detach(), z_H_aug.detach())

    if use_aspect_s:
        d_L = torch.norm(z_L_pert.detach() - z_L.detach(), dim=-1)
        d_H = torch.norm(z_H_pert.detach() - z_H.detach(), dim=-1)
        b_L = norm(loss_L) + lambda_s * norm(d_L)
        b_H = norm(loss_H) + lambda_s * norm(d_H)
```

```
    else:
        b_L = norm(loss_L)
        b_H = norm(loss_H)

    target_logits = torch.stack([-b_L, -b_H], dim=-1) / tau_g
    target_m = softmax(target_logits, dim=-1)[:, 0]

policy_loss = BCE(m_v, target_m)
```

Numerical clipping can be applied to m_v when evaluating the BCE term. The detached target prevents L_{pol} from changing the channel-cost estimates through the pseudo-target. Since the target reuses detached channel representations and is constructed without gradient tracking, it adds little memory overhead and requires no additional encoder forward pass.

Sparse Rayleigh computation. The Rayleigh quotient is computed directly with sparse Laplacian-vector products, without eigendecomposition. For

$$\mathcal{R}(\mathbf{A}, \mathbf{Z}) = \frac{\mathrm{Tr}(\mathbf{Z}^\top \mathbf{L}_\mathbf{A} \mathbf{Z})}{\mathrm{Tr}(\mathbf{Z}^\top \mathbf{Z})},$$

we compute

$$\mathbf{Y} = \mathbf{L}_\mathbf{A} \mathbf{Z}, \qquad \mathcal{R}(\mathbf{A}, \mathbf{Z}) = \frac{\langle \mathbf{Z}, \mathbf{Y} \rangle_F}{\|\mathbf{Z}\|_F^2 + \epsilon}.$$

A code-style implementation is:

```
LZ = sparse_matmul(L_A, Z)
numerator = (Z * LZ).sum()
denominator = (Z * Z).sum()
R = numerator / (denominator + eps)
```

The Rayleigh search term is then computed as

$$\Phi_{\text{ray}} = \mathcal{R}(\mathbf{A}_{\text{pert}}, \mathbf{Z}_L^{\text{pert}}) - \mathcal{R}(\mathbf{A}_{\text{pert}}, \mathbf{Z}_H^{\text{pert}}).$$

For sparse graphs, this computation scales with sparse Laplacian multiplication, approximately $O(|E_{\text{pert}}|d)$, where d is the embedding dimension. This avoids expensive spectral decomposition and makes the Rayleigh spectral search bias practical for large sparse graphs.

These optimizations are engineering choices for scalability. They are used to reduce the time and memory overhead of ASPECT-S while preserving the same training objectives described in Section 3.

E.6 Computational Complexity

Let $n = |\mathcal{V}|$, $m = |\mathcal{E}|$, d be the hidden dimension, and K be the spectral filter order. The dominant cost of ASPECT comes from the dual-channel encoder, whose sparse spectral propagation scales as $O(Kmd)$. The node-wise policy and detached target are node-linear and introduce no additional encoder forward passes. With T_{pert} perturbation steps, perturbation interval r, sparse candidate edge set size m_{cand}, and feature perturbation budget q_f, ASPECT-S adds amortized time $O(\frac{T_{\text{pert}}}{r}(Kmd + m_{\text{cand}} + q_f + md))$; the final md term is for sparse Rayleigh computation. Its additional memory is dominated by perturbed embeddings and sparse perturbation variables, $O(nd + m_{\text{cand}} + q_f)$. Sampled contrastive scoring avoids a dense $O(n^2)$ node-pair similarity matrix on large graphs.

E.7 Model Hyperparameters and Selection Criterion

For ASPECT and all baselines rerun in our pipeline, we tuned hyperparameters using Optuna. For each rerun baseline, the search space was centered around the configuration recommended in the original paper. HLCL results are copied from PolyGCL as noted in Appendix E.2 and are therefore not included in our Optuna tuning pipeline. Final settings were selected by validation accuracy on the clean graph.

Table 11: Hyperparameters used for each dataset

Parameter	Cora	Citeseer	Pubmed	Cornell	Texas	Wisconsin	Actor	Chameleon	Squirrel
Epochs	2000	500	1000	500	500	2000	500	2000	1500
Patience	180	160	40	160	100	20	120	40	140
LR (η)	0.00013	0.00106	0.00011	0.00073	0.00010	0.00214	0.00398	0.00335	0.00121
LR_1 (η_1)	0.00044	0.00357	0.00535	0.00025	0.00486	0.00016	0.00233	0.00228	0.00157
LR_2 (η_2)	0.00915	0.00199	0.00183	0.00295	0.00137	0.00170	0.00054	0.00818	0.00817
LR_α (η_α)	0.14373	26.1982	1.48472	2.63077	0.18482	12.8336	95.5903	12.7409	0.15628
LR_β (η_β)	0.00072	0.00026	0.00124	0.01863	0.00111	0.00051	0.00017	0.00138	0.08001
ϵ	4.05399	1.16728	0.39319	0.83449	1.37270	3.48387	0.66148	3.98710	0.35897
WD (λ)	0.00134	0.00030	0.00786	0.09682	0.00897	3.21e-05	0.09832	0.09787	0.00105
WD_1 (λ_1)	0.00158	0.00356	0.00010	0.00462	0.04208	0.06565	0.01628	0.00018	8.15e-06
WD_2 (λ_2)	0.00202	0.00313	8.34e-05	0.00825	0.09067	0.05710	0.01122	0.00024	2.71e-06
Rayleigh (λ_{ray})	0.46024	0.07248	0.96707	1.19355	1.71332	0.31904	0.08448	0.90943	0.61738
Perturbation Steps	9	5	5	10	4	7	4	7	3
Perturbation Budget	0.22765	0.11267	0.29437	0.12920	0.46972	0.22592	0.45570	0.35284	0.21216
Hidden Dim	512	512	512	512	256	512	512	512	512
K	5	2	4	5	5	5	5	5	5
Dropout	0.34248	0.47064	0.03399	0.45193	0.57931	0.56790	0.04807	0.60798	0.69773
DP Rate	0.45262	0.28825	0.45139	0.72541	0.04969	0.87453	0.04567	0.47966	0.34687
τ	0.26108	0.20047	0.12469	0.69792	0.60886	0.79692	0.27668	0.12598	0.10106
Batch Norm	False	False	True	False	False	False	False	True	True
Activation	prelu	prelu	prelu	prelu	prelu	relu	prelu	relu	prelu

E.8 Hardware and Software Environment

Except for the arXiv-year evaluation described in Appendix C.1, experiments were conducted on the following computing environment:

- **CPU:** AMD EPYC 9554 64-Core Processor @ 3.10GHz (64 Cores, 128 Threads)
- **GPU:** NVIDIA RTX A6000 (48GB GDDR6 memory)
- **RAM:** 256GB DDR4
- **Operating System:** Ubuntu 24.04.2 LTS
- **Python Version:** 3.12.9
- **Deep Learning Framework:** PyTorch 2.4.1
- **GPU Acceleration Libraries:**
  - CUDA Toolkit 12.0
  - cuDNN 9.1.0
- **Other Key Python Libraries:**
  - NumPy 1.26.4
  - SciPy 1.13.1
  - scikit-learn 1.6.1
  - PyTorch Geometric (PyG) 2.6.1 (for graph data structures and operations)

An environment specification listing exact library versions will be provided with the anonymized code package.

F Limitations

Scalability of perturbation generation. Although ASPECT adds only lightweight node-wise policy computation, ASPECT-S requires perturbation generation and therefore introduces additional training cost. We mitigate this cost using sparse perturbation candidates, perturbation intervals, detached policy targets, and sparse Rayleigh computation, but further scaling ASPECT-S to very large graphs remains an important direction.

Scope of spectral views. ASPECT focuses on adaptive fusion between low- and high-frequency views constructed by learnable spectral filters. This design captures a useful two-channel decomposition, but richer multi-band spectral policies or node-dependent filter families may further improve flexibility.

Perturbation model. ASPECT-S uses generated graph-structure and feature perturbations as empirical probes of channel-wise sensitivity. These perturbations are not exact optimizers of the theoretical sensitivity term and may not cover all possible distribution shifts encountered in deployment.

Evaluation scope. Our experiments focus on transductive node classification benchmarks. Extending the analysis to larger-scale graphs, inductive settings, temporal graphs, and other downstream tasks is left for future work.

G BROADER IMPACTS

This work studies self-supervised graph representation learning and is not tied to a specific deployment domain. Its potential positive impacts include improving the quality and stability of graph representations, which may benefit downstream applications such as scientific networks, recommendation systems, and relational data analysis. However, graph representation learning methods can also be used in sensitive domains involving social, behavioral, or user-interaction graphs. In such settings, improved representations may amplify existing biases, enable undesired profiling, or expose privacy-sensitive relational patterns if deployed without appropriate safeguards. Moreover, the perturbation evaluation in this work is limited to controlled structural and feature perturbations and should not be interpreted as a guarantee of security in real-world deployments. We therefore recommend that applications in sensitive domains include domain-specific validation, privacy protection, fairness evaluation, and human oversight.