ADAPTIVE ALIGNMENT-DRIVEN FEATURE AUGMENTATION FOR GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNNs) have achieved strong performance on a wide range of graph learning tasks by integrating node attributes with graph structure through message passing. However, when structural and node attributes encode signals that are weak, inconsistent, or misaligned with the supervision objective, message passing can amplify noise and hinder performance. Existing approaches primarily address these challenges through architectural or structural modifications, and existing feature augmentation methods generally do not account for how the downstream GNN propagates the augmented signals. In this work, we adopt a data-centric perspective and propose GRAPHALFA (ALignment-driven Feature Augmentation), an adaptive, propagation-aware graph feature augmentation framework that explicitly enhances feature-structure-label alignment under task supervision. Rather than modifying model architecture or graph topology, GRAPHALFA operates directly in the feature space by constructing complementary attribute-derived and structure-derived views, and optimizing them through complementary alignment objectives: supervised contrastive learning promotes label-discriminative representations, while a propagation-aware spectral objective estimates the downstream GNN's empirical frequency response and uses it to emphasize task-relevant spectral components effectively preserved by propagation. By conditioning augmentation on this empirical response instead of a fixed spectral prior, GRAPHALFA adapts the input representation to the propagation behavior of the backbone while preserving the graph topology and backbone architecture. Theoretically, we provide an information-theoretic characterization of feature-label alignment and a propagation-aware characterization of spectral feature-structure alignment. Extensive experiments on synthetic and 9 real-world datasets spanning diverse homophily levels show that GRAPHALFA consistently outperforms state-of-the-art baselines, achieving average accuracy gains of 4.82% on real datasets and 12.22% on synthetic datasets over original features.

1 INTRODUCTION

Graphs naturally model relational data in many domains, including social networks (Qiu et al., 2018), molecules (Duvenaud et al., 2015), and recommender systems (Gao et al., 2023). Graph Neural Networks (GNNs) learn on such data by integrating node attributes with graph structure (Kipf & Welling, 2017; Velickovic et al., 2017; Hamilton et al., 2017), achieving strong performance across tasks. The effectiveness of GNNs rests on an implicit data-level assumption: node attributes, graph structure, and task labels should induce mutually consistent notions of similarity. We use *feature-structure-label misalignment* to refer to violations of this assumption, where attribute similarity, structural or role similarity, and label similarity disagree. Such misalignment can appear as noisy or weakly informative features, graph neighborhoods that connect task-dissimilar nodes, or spectral components that are poorly preserved by the downstream model. When this occurs, message passing can amplify noise, suppress informative components, and degrade performance (Arnaiz-Rodríguez & Errica, 2025; Dong et al., 2025). This suggests that some limitations of GNNs are data-driven rather than architectural (Zhang et al., 2022; Guo et al., 2023), and can arise in both homophilic and heterophilic settings rather than being resolved by enforcing homophily (Zheng et al., 2024).

Existing approaches address this primarily on either the model or graph data. Model-centric methods modify aggregation or filtering, including heterophilic GNNs (Zhu et al., 2020) and adaptive spec-

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.

tral models (Zhang et al., 2025). Data-centric approaches modify the graph itself through rewiring or feature augmentation (Ding et al., 2022). Feature augmentation is particularly attractive because it enriches the downstream inputs without altering the observed topology or backbone architecture. However, existing methods primarily determine what structural, positional, or neighborhood information to inject or how to transform it, without explicitly accounting for how GNNs propagate the resulting signals. This creates an important disconnect: augmented information may be task-relevant at the input yet attenuated during downstream propagation, such that augmentation designed without accounting for the GNN can introduce signals that are subsequently suppressed.

We address this disconnect by formulating graph feature augmentation as a *feature-structure-label alignment* problem: effective augmentation should produce representations that are both label-discriminative and compatible with structure-induced propagation, concentrating task-relevant information in graph-frequency regions that GNNs can effectively propagate. Together, these motivate *propagation-aware feature augmentation*. We adapt the input representation to the GNN's propagation behavior, while preserving the observed graph topology and the model architecture.

Building on this principle, we introduce GRAPHALFA (ALignment-driven Feature Augmentation), which constructs complementary attribute- and structure-derived views, projects and adaptively fuses them in a shared latent space, and maps the result to the input space through a residual feature correction. Training combines supervised contrastive learning for feature-label alignment with a propagation-aware spectral objective for feature-structure alignment, which estimates the GNN's empirical frequency response from spectral-energy changes across propagation layers and uses it to weight alignment between augmented feature and label energy profiles, emphasizing task-relevant frequency regions preserved by the backbone. This enables joint adaptation to the graph, task, and backbone without imposing a fixed low-pass, homophily, or smoothness prior.

Our key contributions are summarized as follows:

- **Alignment-driven formulation of feature augmentation.** We formulate graph feature augmentation as feature-structure-label alignment, and propose an adaptive augmentation framework that integrates complementary attribute- and structure-derived information under task supervision while leaving graph topology and downstream backbone unchanged.
- **Propagation-aware spectral feature alignment.** We introduce an empirical-response-guided spectral alignment objective that uses the downstream GNN's observed band-wise propagation behavior to effectively emphasize task-relevant frequencies transmitted by the backbone. We provide a theoretical characterization of this propagation-aware alignment.
- **Extensive empirical validation.** On node classification, we demonstrate consistent performance improvements over state-of-the-art feature augmentation baselines across backbone architectures and nine diverse datasets. Relative to original features, GRAPHALFA achieves average gains of 4.82% on real-world datasets and 12.22% on synthetic datasets.

2 RELATED WORK

Graph Data Augmentation. Data augmentation, extensively studied in computer vision (Cubuk et al., 2019) and natural language processing (Feng et al., 2021b), has been extended to graph-structured data (Ding et al., 2022), by modifying node features or graph structure to improve downstream performance. Feature augmentation is particularly flexible, as it operates directly on potentially noisy or incomplete node attributes, while preserving observed graph structure. Existing approaches can be broadly grouped into predefined and learned. Predefined approaches inject structure-derived signals into node attributes using fixed transformations. For instance, EigenGNN (Zhang et al., 2023) augments node features with Laplacian eigenvectors to encode global structure, NetInfoF (Lee et al., 2024) constructs multiple embeddings capturing structural, attribute, and propagated information. These approaches, however, are not adaptive to task supervision. Learned approaches instead optimize augmentation jointly with downstream objectives, including imputing missing attributes (Taguchi et al., 2021; Chen et al., 2022), improving adversarial robustness (Kong et al., 2020; Feng et al., 2021a), or addressing GNN limitations such as degree bias and the 1-WL bound (Subramonian et al., 2024; Morris et al., 2019) through structural or positional signals. Examples include LLPE (Ito et al., 2025), which learns task-adaptive transformations of

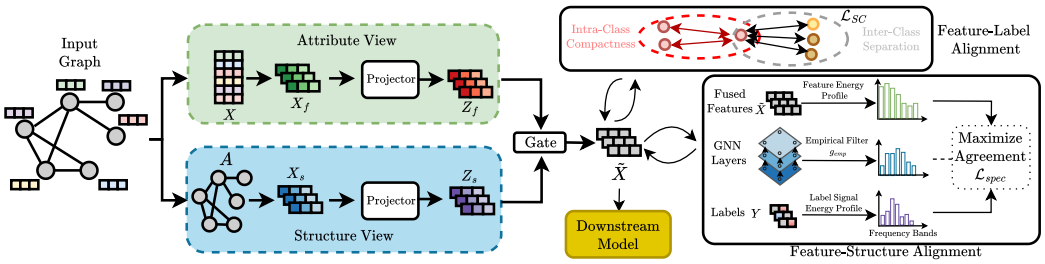


Figure 1: GRAPHALFA is a supervised framework for propagation-aware graph feature augmentation through feature-structure-label alignment. It integrates complementary attribute- and structure-derived views, promotes label-discriminative representations, and uses the downstream GNN's empirical propagation behavior to guide spectral feature alignment.

Laplacian positional encodings; GERF (Bielak et al., 2022), which refines embeddings via neighborhood consistency; and LAGNN (Liu et al., 2022), which generates neighborhood features.

Our work falls within learned feature augmentation, but differs in how downstream propagation guides the augmentation. Prior approaches inject or learn structural or positional information under task supervision, but generally do not condition augmentation on the realized propagation behavior of the downstream GNN. GRAPHALFA instead formulates augmentation as feature-structure-label alignment and uses the GNN's empirical frequency response for spectral alignment, to emphasize task-relevant frequency components. Thus, rather than adapting the propagation operator, GRAPHALFA adapts the input representation to its observed propagation behavior while leaving graph topology and backbone architecture unchanged.

GNNs from a Spectral Perspective. GNNs are grounded in spectral graph theory and graph signal processing (Chung, 1997). Spectral GNNs, such as GCN (Kipf & Welling, 2017), ChebNet (Defferrard et al., 2016), and BernNet (He et al., 2021), interpret graph convolution as filtering graph signals in the frequency domain via functions of the graph Laplacian (Hammond et al., 2009), while spatial GNNs (Velickovic et al., 2017; Hamilton et al., 2017) perform localized neighborhood aggregation. These views are closely related: spatial GNNs can be interpreted as applying implicit spectral filters, enabling spectral analysis (Balcilar et al., 2021). Although GNNs are often characterized as low-pass filters due to aggregation, recent analyses demonstrate that their spectral behavior is task- and supervision-dependent, and can produce diverse frequency components (Dong et al., 2025). This spectral perspective underpins our propagation-aware spectral alignment.

3 METHODOLOGY

We propose GRAPHALFA, a propagation-aware feature augmentation framework that learns feature-structure-label alignment. As shown in Fig. 1, GRAPHALFA constructs complementary node-level views (Sec. 3.1), adaptively fuses them (Sec. 3.2), and optimizes them through complementary feature-label and feature-structure alignment objectives (Sec. 3.3).

Preliminaries Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $N = |\mathcal{V}|$ nodes, node features $\mathbf{X} \in \mathbb{R}^{N \times F}$, adjacency matrix $\mathbf{A}$, and normalized adjacency $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. We consider semi-supervised transductive node classification, where node labels $\mathbf{Y} \in \{1, \ldots, C\}^N$ are available on a training set $\mathcal{V}_{\text{tr}} \subset \mathcal{V}$. Given a downstream backbone, we learn feature augmentation that improves node classification while preserving the graph topology and backbone architecture.

3.1 MULTI-CHANNEL VIEW CONSTRUCTION

Attribute-Derived Context View. In standard message passing, node attributes from different neighborhoods are entangled through aggregation, obscuring the distinction between ego and contextual information. To preserve attribute information at multiple neighborhood scales, we

construct a multi-channel attribute-derived view of explicit ℓ-hop contexts. Formally, we define $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\, \mathbf{X}^{(1)} \,\|\, \mathbf{X}^{(2)}\,]$, where $\mathbf{X}^{(\ell)} = \hat{\mathbf{A}}^{\ell}\mathbf{X}$ and $\mathbf{X}^{(0)} = \mathbf{X}$ denotes ego features. Explicitly separating ego and neighborhood information can improve robustness and expressiveness, while higher-order contexts provide complementary signals, particularly under heterophily (Zhu et al., 2020; Luan et al., 2022). We report ablations for this design in Tables 2 and 8.

Structure-Derived Embedding View. Message-passing GNNs encode structure implicitly through neighborhood aggregation, which is limited in capturing global, positional, or role-based patterns, and is bounded by the expressiveness of the 1-WL test (Morris et al., 2019; Xu et al., 2019). As a result, structural properties such as functional equivalence or community membership may be indistinguishable. We therefore incorporate structure-derived embedding view $\mathbf{X}_s \in \mathbb{R}^{N\times d_s}$, independent of node attributes, to provide complementary structural information. We instantiate it using standard positional and structural embeddings, such as node2vec (Grover & Leskovec, 2016), LINE (Tang et al., 2015), role2vec (Ahmed et al., 2018), and Laplacian positional encodings (You et al., 2019). We compare the effectiveness and scalability of these choices in Apps. D.3 and D.4.

3.2 Projector-Gate Fusion for Adaptive View Alignment

Since the attribute- and structure-derived views encode heterogeneous signals, we first map them into a shared latent space before fusion. Specifically, we use separate learnable projections $\mathbf{Z}_f = f_{\text{attr}}(\mathbf{X}_f)$, $\mathbf{Z}_s = f_{\text{struc}}(\mathbf{X}_s)$, where $f_{\text{attr}} : \mathbb{R}^{d_f} \to \mathbb{R}^{d_p}$ and $f_{\text{struc}} : \mathbb{R}^{d_s} \to \mathbb{R}^{d_p}$. This enables end-to-end alignment before fusion while avoiding early mixing of the heterogeneous input spaces. Since the relative utility of attribute- and structure-derived information varies across nodes and tasks, we use an adaptive node- and dimension-wise gate rather than fixed concatenation or static weighting. Specifically, we compute a gating matrix: $\mathbf{G} = \sigma(\mathbf{Z}_f\mathbf{W}_f + \mathbf{Z}_s\mathbf{W}_s)$, where $\mathbf{W}_f, \mathbf{W}_s \in \mathbb{R}^{d_p\times d_p}$ are learnable parameters and $\sigma(\cdot)$ denotes the element-wise sigmoid. The resulting gate $\mathbf{G} \in [0,1]^{N\times d_p}$ assigns, for each node and each latent dimension, a soft preference between the attribute-derived and structure-derived representations. The fused representation is then $\tilde{\mathbf{X}} = \mathbf{G} \odot \mathbf{Z}_f + (1-\mathbf{G}) \odot \mathbf{Z}_s$. Because the gate is learned end-to-end under task supervision, the fusion adapts automatically to the data and task, providing fine-grained control over information flow. Finally, we inject the fused representation through a residual update: $\mathbf{X}_{\text{in}} = \mathbf{X}_{\text{base}} + \Delta\mathbf{X}_\theta$, $\Delta\mathbf{X}_\theta = r_\theta(\tilde{\mathbf{X}})$, where $\mathbf{X}_{\text{base}}$ denotes the backbone input, and r_θ is a learnable residual mapping. The residual form preserves baseline recoverability, since $\Delta\mathbf{X}_\theta = \mathbf{0}$ exactly recovers the original input. We further regularize the augmentation magnitude via $\|\Delta\mathbf{X}_\theta\|_F^2$.

3.3 Supervised Alignment-driven Objectives

The fused representation $\tilde{\mathbf{X}}$ is optimized end-to-end using alignment-driven objectives that explicitly couple features, structure, and task supervision. While the projector-gate module enables flexible integration of attribute-derived and structure-derived information, fusion alone does not specify which shared signals are task-relevant (i.e., aligned with the task). We therefore impose two complementary objectives: (i) a feature-label alignment objective that enforces intra-class similarity and inter-class separation in the augmented feature space, and (ii) a feature-structure alignment objective that aligns the spectral energy of the augmented features with that of the task labels, weighting each frequency band according to how strongly it is preserved by the downstream GNN propagation.

3.3.1 Feature-Label Alignment.

While the projector-gate fusion increases representational capacity, it does not guarantee that the augmented features are discriminative with respect to task labels. In particular, fused representations may encode additional information without exhibiting clear intra-class compactness or inter-class separation. To explicitly align the augmented feature space with task supervision, we impose a supervised contrastive objective on the fused representations. Given a mini-batch $\mathcal{I}$ of nodes with fused representations $\tilde{\mathbf{x}}_i$, we define the contrastive loss $\mathcal{L}_{\text{SC}}$ as:

$$-\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\frac{1}{|\mathcal{P}(i)|}\sum_{j\in\mathcal{P}(i)}\log\frac{\exp\big(\tilde{\mathbf{x}}_i^\top\tilde{\mathbf{x}}_j/\tau\big)}{\sum_{v\in\mathcal{I}\setminus\{i\}}\exp\big(\tilde{\mathbf{x}}_i^\top\tilde{\mathbf{x}}_v/\tau\big)},$$

where $\mathcal{P}(i)$ is the set of nodes in $\mathcal{I}$ sharing the same class label as node i, and τ is a temperature parameter. This objective enforces intra-class compactness and inter-class separation in the augmented feature space, shaping the geometry of $\tilde{\mathbf{X}}$ such that feature similarity reflects label similarity.

3.3.2 PROPAGATION-AWARE FEATURE-STRUCTURE ALIGNMENT.

Even when augmented features are well aligned with task labels, their effectiveness depends on how task-relevant information is propagated over the structure by the model. Since message passing implicitly acts as a graph filter that selectively preserves or suppresses different Laplacian frequencies, we introduce a propagation-aware feature-structure alignment objective that aligns the augmented features with the GNN's structure-induced propagation of supervision signals in the spectral domain. Unlike fixed homophily or smoothness priors, it targets frequency bands that are both label-informative and preserved by the chosen backbone.

To support this alignment between features and how task supervision propagates over the structure, we adopt a band-wise spectral representation that provides a common abstraction for (i) estimating the empirical filtering behavior of the GNN, (ii) characterizing the spectral energy distributions of features and labels, and (iii) defining a propagation-aware alignment objective. We discretize the normalized Laplacian spectrum $[0, 2]$ into K disjoint frequency bands $\{\mathcal{B}_k\}_{k=1}^{K}$. For each band $\mathcal{B}_k$, we define the spectral mask $\mathbf{M}_k := \mathrm{diag}(\mathbf{1}[\lambda_i \in \mathcal{B}_k])_{i=0}^{N-1}$ and the corresponding node-domain band-pass projector $\mathbf{\Pi}_k = \mathbf{U}\mathbf{M}_k\mathbf{U}^\top$, which isolates signal components in the corresponding frequency range. In practice, $\mathbf{\Pi}_k$ is approximated via stochastic spectral slicing.

Spectral Energy Profiles of Features and Labels. To enable spectral alignment between augmented node features $\tilde{\mathbf{X}}$ and task supervision $\mathbf{Y}_{\mathrm{tr}}$, we characterize how both signals are distributed across the graph Laplacian spectrum using a shared band-wise spectral energy representation, computed efficiently via the same stochastic slicing procedure. For a signal $\mathbf{S} \in \mathbb{R}^{N\times d}$, the spectral energy in frequency band $\mathcal{B}_k$ is defined $E_{\mathbf{S}}(k) = \|\mathbf{\Pi}_k\mathbf{S}\|_F^2$, where $\mathbf{\Pi}_k$ is the band-pass projector. Applying this to the augmented features $\tilde{\mathbf{X}}$ and the train-masked one-hot label matrix $\mathbf{Y}_{\mathrm{tr}} \in \mathbb{R}^{N\times C}$ yields feature energy profile $\mathbf{E}_f \in \mathbb{R}^K$, where each entry quantifies the amount of feature energy in the corresponding frequency band, and label energy profile $\mathbf{E}_y \in \mathbb{R}^K$, which characterizes how task supervision is distributed across frequency bands.

Empirical GNN Filter Estimation. While GNN message passing can be analyzed through a spectral filtering lens, the effective frequency behavior of a trained GNN is generally not given by a fixed analytic filter. In particular, nonlinearities, normalization, attention weights, and learned parameters can make the propagation behavior data- and model-dependent, not necessarily low-pass (Dong et al., 2025), and may differ substantially from theoretical interpretations. We therefore empirically estimate the effective spectral propagation behavior by measuring how spectral energy is redistributed across layers. Let $\mathbf{H}^{(\ell)}$ and $\mathbf{H}^{(\ell+1)}$ denote the node representations before and after layer ℓ. To characterize how a GNN layer propagates information across the graph spectrum, we estimate the redistribution of band-wise spectral energy induced by the layer. We use stochastic Lanczos-based spectral slicing (Levie et al., 2019; Ubaru et al., 2017) to approximate band-pass projections in a low-dimensional Krylov subspace of the graph Laplacian. For each frequency band, we compute the ratio of spectral energy in $\mathbf{H}^{(\ell+1)}$ to that in $\mathbf{H}^{(\ell)}$, which estimates how strongly that band is preserved by the observed layer transformation. Averaging these band-wise ratios across layers yields an empirical frequency response $\mathbf{g}_{\mathrm{emp}} \in \mathbb{R}^K$, capturing the extent to which different spectral components are preserved. This provides a data- and model-specific estimate and is used to weight the feature-structure alignment.

Backbone-weighted Spectral Alignment. We define a spectral alignment objective that encourages the augmented features to match the relative spectral energy distribution of the task supervision across frequency bands, while remaining invariant to energy scale. Concretely, we align the feature and label spectral energy profiles using a weighted cosine similarity loss:

$$\mathcal{L}_{\mathrm{spec}} = 1 - \frac{\sum_{k=1}^{K} w_k\, E_{f,k}\, E_{y,k}}{\sqrt{\left(\sum_{k=1}^{K} w_k\, E_{f,k}^2\right)\left(\sum_{k=1}^{K} w_k\, E_{y,k}^2\right)}}. \tag{1}$$

The weights are obtained by normalizing the estimated frequency response, $w_k = g_{\mathrm{emp},k} / \sum_{i=1}^{K} g_{\mathrm{emp},i}$, assigning more importance to frequency bands more preserved by the message passing. By incorporating the empirically estimated propagation bias, our alignment guides augmented features toward task-relevant spectral components that are effectively utilized by the downstream GNN. Unlike fixed homophily or smoothness priors, it can emphasize lower- or higher-frequency components depending on the data and task, thereby adapting jointly to the data and model.

4 Theoretical Analysis of GraphALFA

We theoretically characterize the two alignment objectives underlying GraphALFA. Specifically, contrastive alignment provides an information-theoretic characterization of feature-label coupling, while spectral alignment admits a propagation-aware interpretation of feature-structure matching. Let f_φ denote a GNN backbone mapping $(\mathbf{A}, \mathbf{X})$ to node predictions. GraphALFA augments its input with a learned residual update $\Delta\mathbf{X}_\theta(\mathbf{A}, \mathbf{X})$, and applies the backbone to the resulting augmented features.

4.1 Feature-Label Information Coupling via Contrastive Alignment

Proposition 1 (Feature-label alignment via contrastive learning)**.** *Let $Z = \tilde{\mathbf{x}}_i \in \mathbb{R}^{d_p}$ denote the fused representation of node i, Y its class label, and Z^+ a positive sample drawn from the same class. Consider supervised contrastive training with mini-batches $\mathcal{I} \subseteq \mathcal{V}_{\mathrm{tr}}$ of size $|\mathcal{I}|$. Define the population SupCon objective $\bar{\mathcal{L}}_{\mathrm{SC}} := \mathbb{E}\big[\mathcal{L}_{\mathrm{SC}}(\mathcal{I})\big]$, where the expectation is over the data distribution and the mini-batch/positive/negative sampling used to form $\mathcal{I}$. Then: $I(Z;Y) \geq I(Z;Z^+) \gtrsim \log(|\mathcal{I}|-1) - \bar{\mathcal{L}}_{\mathrm{SC}}$.*

Thus, for fixed batch size, minimizing $\bar{\mathcal{L}}_{\mathrm{SC}}$ increases an InfoNCE-style variational lower bound linked to $I(Z;Y)$ through the DPI relation $I(Z;Y) \geq I(Z;Z^+)$, under the standard sampling and critic assumptions. This provides an information-theoretic characterization of how supervised contrastive learning promotes feature-label alignment. Derivations are provided in App. B.4.

4.2 Propagation-Aware Spectral Matching for Feature-Structure Alignment

Feature-structure alignment is motivated by a *propagation-aware* perspective: label signals concentrate in specific frequency regions, while GNN propagation transmits these frequencies in a model- and graph-dependent manner (Zhang et al., 2025; Dong et al., 2025), which is often non-monotone and homophily-dependent. Consequently, feature alignment should account not only for where task-relevant spectral information is concentrated, but also for how strongly those frequency components are preserved by the downstream backbone. We characterize this principle through a backbone-weighted spectral alignment objective.

Spectral characterization of propagation. We partition the spectrum into bands and define band energies $E_{\mathbf{S}}(k)$ and profiles $\mathbf{E}_{\mathbf{S}}$, then analyze how message passing redistributes spectral energy via a spectral-filter approximation: $\mathbf{T} = \mathbf{U}\,\mathrm{diag}\big(t(\lambda_0), \ldots, t(\lambda_{N-1})\big)\,\mathbf{U}^\top$, where $t(\lambda)$ is the backbone's effective spectral response at eigenvalue λ, i.e., the scalar by which the λ-frequency component of a graph signal is scaled under propagation.

Lemma 1 (Band-wise propagation preservation)**.** *Let $\mathbf{T}$ denote a spectral-filter approximation of a GNN. For any graph signal $\mathbf{S} \in \mathbb{R}^{N\times d}$, let $\mathbf{E}_{\mathbf{S}} \in \mathbb{R}^K_{\geq 0}$ denote its band-wise spectral energy profile. For each frequency band $\mathcal{B}_k$, define $t_{\min,k} = \min_{i:\,\lambda_i\in\mathcal{B}_k} |t(\lambda_i)|$ and $t_{\max,k} = \max_{i:\,\lambda_i\in\mathcal{B}_k} |t(\lambda_i)|$. Then, the spectral energy of the propagated signal in band $\mathcal{B}_k$, $E_{\mathbf{TS}}(k) := \|\mathbf{\Pi}_k(\mathbf{TS})\|_F^2$, is bounded as: $t^2_{\min,k}\, E_{\mathbf{S}}(k) \;\leq\; E_{\mathbf{TS}}(k) \;\leq\; t^2_{\max,k}\, E_{\mathbf{S}}(k)$*

We provide the proof in App. B.5. The lemma shows that message passing scales spectral energy in a band-wise manner, with different frequency regions preserved according to the spectral response $t(\lambda)$. This motivates weighting $\mathcal{L}_{\mathrm{spec}}$ by the normalized empirical response $\mathbf{w} \propto \mathbf{g}_{\mathrm{emp}}$. Proposition 2 in App. B.5 further shows that minimizing $\mathcal{L}_{\mathrm{spec}}$ is equivalent to reducing the distance between normalized, backbone-weighted spectral energy profiles of the augmented features and la-

bels. Together, these results characterize $\mathcal{L}_{\text{spec}}$ as propagation-aware feature-structure matching that emphasizes task-relevant frequency regions according to the downstream GNN's propagation.

5 Empirical Analysis

We present an extensive empirical evaluation of GraphALFA. Our analysis aims to answer: (**RQ1**) How effective and generalizable is the proposed GraphALFA, which enhances feature-structure-label alignment, compared to state-of-the-art feature augmentation methods? (**RQ2**) How do the different components and design choices of GraphALFA contribute to the overall effectiveness? (**RQ3**) How does the runtime of GraphALFA compare to that of prior graph feature augmentation methods?

5.1 Empirical Setup

Datasets. We evaluate GraphALFA on 9 real-world node classification datasets, spanning different homophily levels, sizes, and domains: (i) citation graphs (Yang et al., 2016; Lim et al., 2021): Cora, Citeseer, PubMed, Arxiv-Year; (ii) web graphs (Platonov et al., 2023; Mernyei & Cangea, 2020): Squirrel, Chameleon, WikiCS; (iii) social network: Flickr (Zhou et al., 2023); (iv) co-purchase graph: Ratings (Platonov et al., 2023). We give details in App. D.1.1. Additionally, we evaluate GraphALFA on two multimodal datasets (Zhu et al., 2025), Ele-Fashion and Goodreads-NC, in App. D.1.2. Based on standard global homophily measures (Tab. 4), we refer to Chameleon, Squirrel, Flickr, Ratings, and Arxiv-Year as heterophilic data, and all remaining datasets as homophilic.

To enable analysis across graphs of diverse characteristics, we also use Contextual Stochastic Block Model (cSBM) (Deshpande et al., 2018) synthetic graphs, which allow us to control the homophily level. We parameterize the generation process using ϕ_{cSBM}, which controls the balance between structural and feature signals. As ϕ_{cSBM} varies from -1 to 1, the graph varies from strongly heterophilic to strongly homophilic. We give details in App. D.2.1.

Baselines & Backbones. We compare GraphALFA against recent, representative SOTA feature augmentation *baselines*: (1) Rule-based methods: Eigen-GNN (Zhang et al., 2023) and NetInfoF (Lee et al., 2024); (2) Learned methods: LLPE (Ito et al., 2025), GERF (Bielak et al., 2022), and LAGNN (Liu et al., 2022). We evaluate GraphALFA and the baselines across representative *backbones*: GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2017), Adaptive Channel Mixing GNN (ACM-GCN) (Luan et al., 2022), with MLP as a graph-agnostic control.

Metrics and Evaluation Protocols. We report node classification accuracy as the mean and standard deviation on the test set across 5 random splits. For baseline augmentation methods, we use official implementations and the recommended hyperparameter settings. On downstream classification backbone models, we apply the same hyperparameter search setup across all methods to ensure a fair comparison (details in App. D.5). Unless otherwise noted, GraphALFA uses LINE for the structure-derived view $\mathbf{X}_{\mathbf{s}}$.

5.2 Empirical Results

5.2.1 (RQ1) Overall Performance and Generalizability.

We evaluate GraphALFA's performance and generalizability across backbone architectures, comparing against state-of-the-art baselines on both real-world and synthetic datasets.

Real-world Datasets. We summarize classification accuracy (higher is better) and average rank (lower is better) on real-world datasets in Table 1, where *"Original"* denotes backbones without feature augmentation. Overall, GraphALFA achieves the best average rank across all backbones, on both homophilic and heterophilic datasets, indicating that alignment-driven feature augmentation generalizes well. We observe similar trends on ACM-GCN and LLM-enhanced features (Tabs 5 and 6). For GNN backbones, improvements are generally larger on heterophilic graphs, where neighborhood aggregation is more prone to mixing conflicting signals (Zhu et al., 2020). Results on the heterophily-aware ACM-GCN in Table 5 further show this beyond standard GNNs.

Synthetic Datasets. Figure 2 summarizes results on cSBM synthetic graphs across homophily levels

Table 1: Accuracy (%) on real-world datasets: mean accuracy ± stdev across runs for different backbones. Best results in bold. GRAPHALFA achieves the best overall performance.

Dataset	Original	Feature Augmentation					
		EigGNN	NetInfoF	GERF	LLPE	LAGNN	Ours
MLP							
Chamel.	43.15±3.6	44.04±2.2	44.04±3.3	40.11±4.2	43.61±3.7	36.40±3.6	**48.13**±1.3
Squirrel	37.84±0.7	38.43±1.2	36.01±1.9	36.59±1.7	40.51±0.6	34.93±1.1	**43.57**±2.3
Flickr	90.96±0.6	92.12±0.2	89.47±0.8	56.02±0.5	90.69±0.3	91.60±0.7	**94.04**±0.6
Ratings	52.04±0.2	52.25±0.9	54.66±1.3	48.26±1.1	49.14±0.1	40.46±1.5	**57.38**±0.3
Arxiv-Y	37.16±0.5	38.47±0.2	**49.89**±0.3	42.10±0.2	45.66±0.5	29.08±0.8	49.43±0.4
Cora	75.87±1.6	81.88±1.6	86.34±0.7	86.78±1.1	74.89±0.2	81.14±2.1	**89.20**±0.9
PubMed	88.50±0.4	**89.38**±0.4	88.53±0.4	84.86±1.0	87.52±0.3	88.74±0.7	89.00±0.3
WikiCS	81.76±1.0	80.55±1.2	85.63±0.5	81.95±0.5	81.83±0.6	71.16±0.9	**85.89**±0.5
Citeseer	73.36±1.5	74.70±0.9	72.75±1.7	71.99±0.9	73.52±2.0	74.40±1.1	**78.74**±0.3
Avg. Rank		3.0	3.2	4.6	4.0	4.9	**1.2**
GCN							
Chamel.	42.65±5.5	45.51±3.5	45.32±8.8	44.94±5.2	45.13±2.8	44.80±3.9	**48.69**±1.3
Squirrel	39.91±2.1	43.05±1.6	42.53±0.9	44.53±2.0	43.20±2.8	42.94±1.8	**45.81**±3.0
Flickr	92.18±0.4	69.90±2.0	74.94±1.0	62.01±0.6	91.82±0.2	90.26±1.0	**93.53**±0.7
Ratings	49.96±1.3	54.47±1.7	47.07±0.3	51.19±0.9	44.23±1.8	45.17±3.3	**55.41**±0.9
Arxiv-Y	44.70±0.4	45.82±0.3	48.40±0.7	49.22±0.2	46.00±0.4	41.86±0.6	**53.24**±0.1
Cora	87.85±0.3	87.77±0.4	86.49±1.7	87.55±1.4	**88.95**±1.0	88.12±1.8	88.28±1.4
PubMed	88.35±0.7	88.64±0.5	87.75±0.3	84.87±0.7	87.57±0.4	**89.61**±0.1	88.88±0.3
WikiCS	84.08±1.2	84.47±0.5	83.12±0.6	82.56±0.5	83.90±0.5	81.66±0.7	**84.94**±0.7
Citeseer	76.71±0.3	74.78±0.8	73.06±0.5	72.21±2.1	77.83±1.1	76.83±1.0	**77.88**±1.3
Avg. Rank		3.4	4.3	4.4	3.3	4.2	**1.2**
GAT							
Chamel.	41.46±4.9	43.48±3.9	41.91±3.4	40.45±6.3	45.17±4.4	39.78±1.4	**48.88**±2.0
Squirrel	36.55±1.9	37.09±3.0	39.24±3.2	35.47±3.1	46.19±2.6	35.02±2.6	**46.41**±1.9
Flickr	90.60±0.4	84.82±1.2	88.11±0.8	43.45±1.5	90.86±0.4	88.17±1.4	**93.07**±0.9
Ratings	48.39±0.4	50.01±1.1	53.93±0.3	50.06±0.6	41.02±5.2	42.94±4.8	**55.72**±1.7
Arxiv-Y	42.72±0.3	43.17±0.6	51.98±0.4	45.16±0.3	49.31±3.4	38.23±3.0	**52.75**±0.3
Cora	85.86±0.7	86.81±1.6	87.48±1.1	86.34±2.0	87.11±0.7	87.55±1.8	**89.93**±1.0
PubMed	87.74±0.3	88.94±0.4	87.97±0.6	83.73±0.4	85.98±0.7	88.87±0.3	**89.18**±0.8
WikiCS	83.44±1.1	83.05±0.4	83.72±0.5	81.86±0.4	83.40±0.9	81.91±0.9	**85.36**±0.4
Citeseer	73.73±2.1	76.68±1.2	73.36±1.4	70.81±1.6	76.96±1.1	76.56±1.5	**77.98**±1.5
Avg. Rank		3.9	3.2	5.2	3.2	4.4	**1.0**

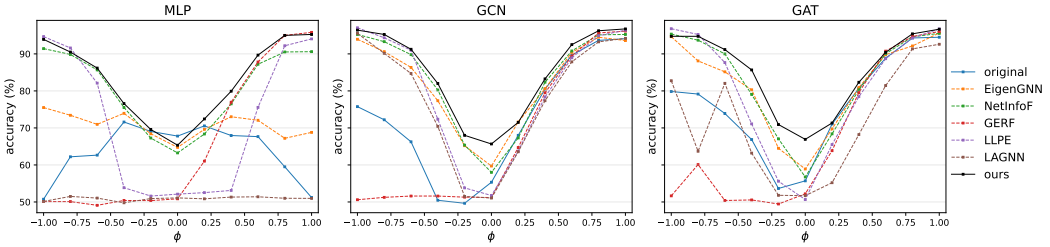


Figure 2: Node classification accuracy (%) on synthetic cSBM graph datasets, under varying homophily levels (x-axis). GRAPHALFA achieves the best overall accuracy.

ϕ_{cSBM}. For GCN and GAT, gains are largest under heterophily, where neighborhood aggregation is more susceptible to conflicting signals, and smaller under strong homophily, as standard message passing already captures informative structure (Zheng et al., 2022). For MLP, gains are minimal when $\phi_{\text{cSBM}} \approx 0$ and structure is nearly label-independent, but improves performance more as $|\phi_{\text{cSBM}}|$ grows and structural signal strengthens under both heterophily and homophily.

5.2.2 (RQ2) ABLATION STUDY.

We perform ablation studies by systematically removing key modules. Table 2 reports results on GCN, with additional backbones in Table 8. The full framework achieves the best performance over-

Table 2: Ablation of GRAPHALFA on GCN, when removing individual components.

Config	Chameleon	Squirrel	Ratings	Cora	Citeseer
Original (no augmentation)	42.65±5.5	39.91±2.1	49.96±1.3	87.85±0.3	76.71±0.3
Concat $[\mathbf{X}_f \,\|\|\, \mathbf{X}_s]$ (no align.)	42.13±4.0	42.56±2.6	46.50±0.8	86.96±1.4	74.98±2.4
No struct view $\mathbf{X}_s$	44.04±2.3	41.08±2.4	37.24±0.6	87.11±1.8	74.43±1.2
Only Ego Feat., $\mathbf{X}_f = \mathbf{X}^{(0)}$	45.88±5.5	43.20±2.3	46.00±1.4	87.85±2.1	77.27±0.6
No 1-hop, $\mathbf{X}_f = [\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(2)}]$	45.51±4.2	43.14±3.4	48.60±0.5	88.18±2.9	75.92±1.0
No 2-hop, $\mathbf{X}_f = [\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(1)}]$	45.06±3.5	44.35±4.2	54.57±1.1	87.33±1.0	77.32±1.0
No feat-label align., only $\mathcal{L}_{\text{spec}}$	43.60±2.9	44.48±1.0	52.61±0.7	87.59±0.8	76.10±1.3
No feat-struct align., only $\mathcal{L}_{\text{SC}}$	44.16±4.7	44.17±2.3	52.96±0.9	87.77±1.7	75.89±1.3
Full Framework	**48.69**±1.3	**45.81**±3.0	**55.41**±0.9	**88.28**±1.4	**77.88**±1.3

Table 3: Runtime overhead (in seconds) of graph feature augmentation methods with a GCN backbone: reported times include additional pre-processing and training time introduced by each method. Lower is better; best runtime is shown in bold. GRAPHALFA has the lowest overall runtime.

Dataset	EigGNN	NetInfoF	GERF	LLPE	LAGNN	Ours
Chamel.	4.2	9.7	255.3	38.4	2.6	**0.9**
Squirrel	23.2	14.3	548.7	228.6	7.6	**1.5**
Flickr	43.0	58.3	2826.9	5102.6	72.0	**10.6**
Ratings	23.6	112.4	5525.0	1844.8	149.7	**8.9**
Arxiv-Y	2769	**62**	11294	21262	367	175
Cora	22.2	22.5	137.5	27.3	4.5	**2.9**
PubMed	133.2	60.9	1116.2	265.5	450.1	**16.7**
WikiCS	53.9	17.4	2329.4	1952.8	277.5	**17.0**
Citeseer	20.4	22.4	164.6	27.0	6.7	**1.8**

all, indicating that components provide complementary gains. Removing the structure-derived view leads to an average drop of 6.43% (9.18% on heterophilic graphs). Removing alignment objectives also noticeably degrades performance (2.28% on average), together with the weaker concatenation variant, this shows that gains depend on both structural augmentation and explicit alignment. These trends are consistent across backbones and datasets. App. D.3 further shows stable performance across structure-derived views, with LINE providing a favorable accuracy-efficiency trade-off, as well as robustness to hyperparameters and noise.

5.2.3 (RQ3) RUNTIME.

Table 3 reports runtime overhead of feature augmentation methods. Overall, GRAPHALFA incurs the lowest or near-lowest runtime overhead across datasets, with average speedups of 7.1x to 217.6x over feature augmentation baselines. This efficiency stems from operating in the feature space, avoiding additional backbone training, and leveraging lightweight, scalable structure-based node embeddings. App. B.1 includes a detailed time and memory complexity analysis. Notably, the dominant graph-dependent components scale linearly with graph size.

6 CONCLUSION

In this work, we proposed GRAPHALFA, a propagation-aware feature augmentation framework that formulates graph feature augmentation as feature-structure-label alignment. Its central mechanism uses the empirically estimated frequency response of the downstream GNN to guide spectral feature alignment, adapting augmented representations toward task-relevant components that are effectively preserved by propagation. Across diverse datasets and backbone architectures, GRAPHALFA consistently improves node classification over strong feature augmentation baselines while maintaining low computational overhead. More broadly, this suggests that effective graph feature augmentation should account not only for what information is introduced, but also for whether that information can be effectively transmitted by downstream propagation. By design, GRAPHALFA is most effective when meaningful structural signals are present. The current formulation targets the standard transductive node classification setting. A promising direction for future work is to integrate GRAPHALFA with graph rewiring methods, which is conceptually complementary and may yield synergistic gains, motivating joint optimization strategies.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research ideation and development, drafting and polishing portions of the manuscript, and developing and checking mathematical arguments and proofs. All AI-assisted work was independently reviewed and verified by the authors. Generative AI was not used to generate synthetic datasets or to source other information used. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed framework and training objectives in the main paper, with end-to-end pseudocode in App. C. Complete theoretical derivations and proofs are provided in App. B. Dataset statistics, processing, and synthetic-data generation are documented in App. D.1.1 and D.2.1. Implementation and parameter details are given in App. D.5.

REFERENCES

Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *J. Mach. Learn. Res.*, 18:177:1–177:86, 2017.

Nesreen K. Ahmed, Ryan A. Rossi, Theodore L. Willke, and Rong Zhou. Edge role discovery via higher-order structures. In *PAKDD*, volume 10234 of *Lecture Notes in Computer Science*, pp. 291–303, 2017.

Nesreen K. Ahmed, Ryan A. Rossi, John Boaz Lee, Xiangnan Kong, Theodore L. Willke, Rong Zhou, and Hoda Eldardiry. Learning role-based graph embeddings. *CoRR*, abs/1802.02896, 2018.

Adrián Arnaiz-Rodríguez and Federico Errica. Oversmoothing, "oversquashing", heterophily, long-range, and more: Demystifying common beliefs in graph machine learning. *CoRR*, abs/2505.15547, 2025.

Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the expressive power of graph neural networks in a spectral perspective. In *ICLR*, 2021.

Luc Bégin, Pascal Germain, François Laviolette, and Jean-Francis Roy. Pac-bayesian theory for transductive learning. In *AISTATS*, 2014.

Piotr Bielak, Daria Puchalska, and Tomasz Kajdanowicz. Retrofitting structural graph embeddings with node attribute information. In *ICCS*, volume 13350 of *Lecture Notes in Computer Science*, pp. 178–191, 2022.

Sandro Cavallari, Erik Cambria, Hongyun Cai, Kevin Chen-Chuan Chang, and Vincent W. Zheng. Embedding both finite and infinite communities on graphs [application notes]. *IEEE Comput. Intell. Mag.*, 14(3):39–50, 2019.

Xu Chen, Siheng Chen, Jiangchao Yao, Huangjie Zheng, Ya Zhang, and Ivor W. Tsang. Learning on attribute-missing graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(2):740–757, 2022.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *ICLR*, 2021.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 113–123, 2019.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pp. 3837–3845, 2016.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. In *NeurIPS*, pp. 8590–8602, 2018.

Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data augmentation for deep graph learning: A survey. *ACM SIGKDD Explorations Newsletter*, 24(2):61–77, 2022.

Yushun Dong, Patrick Soga, Yinhan He, Song Wang, and Jundong Li. Graph neural networks are more than filters: Revisiting and benchmarking from A spectral perspective. In *ICLR*, 2025.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Trans. Knowl. Data Eng.*, 33(6):2493–2504, 2021a.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. A survey of data augmentation approaches for NLP. In *Findings of ACL/IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pp. 968–988, 2021b.

Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD*, pp. 855–864, 2016.

Zhichun Guo, William Shiao, Shichang Zhang, Yozen Liu, Nitesh V. Chawla, Neil Shah, and Tong Zhao. Linkless link prediction via relational distillation. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12012–12033, 2023.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pp. 1024–1034, 2017.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *CoRR*, abs/0912.3848, 2009.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*, pp. 14239–14251, 2021.

Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. Rolx: structural role extraction & mining in large graphs. In *KDD*, pp. 1231–1239, 2012.

Michael Ito, Jiong Zhu, Dexiong Chen, Danai Koutra, and Jenna Wiens. Learning laplacian positional encodings for heterophilous graphs. In *AISTATS*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2755–2763, 2025.

Di Jin, Mark Heimann, Ryan A. Rossi, and Danai Koutra. node2bits: Compact time- and attribute-aware node representations for user stitching. In *ECML/PKDD*, volume 11906 of *Lecture Notes in Computer Science*, pp. 483–506, 2019a.

Di Jin, Mark Heimann, Tara Safavi, Mengdi Wang, Wei Lee, Lindsay Snider, and Danai Koutra. Smart roles: Inferring professional roles in email networks. In *KDD*, pp. 2923–2933, 2019b.

Prannay Khosla, Philip Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. FLAG: adversarial data augmentation for graph neural networks. *CoRR*, abs/2010.09891, 2020.

John Boaz Lee, Ryan A. Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. Higher-order graph convolutional networks. *CoRR*, abs/1809.07697, 2018.

Meng-Chieh Lee, Haiyang Yu, Jian Zhang, Vassilis N. Ioannidis, Xiang Song, Soji Adeshina, Da Zheng, and Christos Faloutsos. Netinfof framework: Measuring and exploiting network usable information. In *ICLR*, 2024.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Trans. Signal Process.*, 67 (1):97–109, 2019.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *NeurIPS*, pp. 20887–20902, 2021.

Jonas Linkerhägner, Cheng Shi, and Ivan Dokmanic. Joint graph rewiring and feature denoising via spectral resonance. In *ICLR*, 2025.

Songtao Liu, Rex Ying, Hanze Dong, Lanqing Li, Tingyang Xu, Yu Rong, Peilin Zhao, Junzhou Huang, and Dinghao Wu. Local augmentation for graph neural networks. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14054–14072, 2022.

Xi Liu, Muhe Xie, Xidao Wen, Rui Chen, Yong Ge, Nick Duffield, and Na Wang. A semi-supervised and inductive embedding model for churn prediction of large-scale mobile games. In *ICDM*, pp. 277–286, 2018.

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 *NeurIPS*, 2022.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Classic gnns are strong baselines: Reassessing gnns for node classification. In *NeurIPS*, 2024.

Péter Mernyei and Catalina Cangea. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *CoRR*, abs/2007.02901, 2020.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *AAAI/IAAI/EAAI*, pp. 4602–4609, 2019.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *ICLR*, 2023.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *KDD*, pp. 2110–2119, 2018.

Ryan A. Rossi, Rong Zhou, and Nesreen K. Ahmed. Deep feature learning for graphs. *CoRR*, abs/1704.08829, 2017.

Ryan A. Rossi, Nesreen K. Ahmed, and Eunyee Koh. HONE: higher-order network embeddings. *CoRR*, abs/1801.09303, 2018.

Ryan A. Rossi, Di Jin, Sungchul Kim, Nesreen K. Ahmed, Danai Koutra, and John Boaz Lee. On proximity and structural role-based embeddings in networks: Misconceptions, techniques, and applications. *ACM Trans. Knowl. Discov. Data*, 14(5):63:1–63:37, 2020.

Arjun Subramonian, Jian Kang, and Yizhou Sun. Theoretical and empirical insights into the origins of degree bias in graph neural networks. In *NeurIPS*, 2024.

Hibiki Taguchi, Xin Liu, and Tsuyoshi Murata. Graph convolutional networks for graphs containing missing features. *Future Gener. Comput. Syst.*, 117:155–168, 2021.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. In *WWW*, pp. 1067–1077, 2015.

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of tr(f(a)) via stochastic lanczos quadrature. *SIAM J. Matrix Anal. Appl.*, 38(4):1075–1099, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *NeurIPS*, 2018.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. *CoRR*, abs/1710.10903, 2017.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 40–48, 2016.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7134–7143, 2019.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. In *ICLR*, 2022.

Shuaicheng Zhang, Haohui Wang, Junhong Lin, Xiaojie Guo, Yada Zhu, Si Zhang, Dongqi Fu, and Dawei Zhou. Herofilter: Adaptive spectral graph filter for varying heterophilic relations. In *NeurIPS*, volume 38, pp. 64537–64559, 2025.

Ziwei Zhang, Peng Cui, Jian Pei, Xin Wang, and Wenwu Zhu. Eigen-gnn: A graph structure preserving plug-in for gnns. *IEEE Trans. Knowl. Data Eng.*, 35(3):2544–2555, 2023.

Xin Zheng, Yixin Liu, Shirui Pan, Miao Zhang, Di Jin, and Philip S. Yu. Graph neural networks for graphs with heterophily: A survey. *CoRR*, abs/2202.07082, 2022.

Yilun Zheng, Sitao Luan, and Lihui Chen. What is missing in homophily? disentangling graph homophily for graph neural networks. *arXiv preprint arXiv:2406.18854*, 2024.

Zhiyao Zhou, Sheng Zhou, Bochao Mao, Xuanyi Zhou, Jiawei Chen, Qiaoyu Tan, Daochen Zha, Yan Feng, Chun Chen, and Can Wang. Opengsl: A comprehensive benchmark for graph structure learning. In *NeurIPS*, 2023.

Jing Zhu, Yuhang Zhou, Shengyi Qian, Zhongmou He, Tong Zhao, Neil Shah, and Danai Koutra. Mosaic of modalities: A comprehensive benchmark for multimodal graph learning. In *CVPR*, pp. 14215–14224, 2025.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020.

A Related Work

Node Embeddings. Node embeddings are node representations derived solely from graph topology, aiming to capture structural similarity or functional roles (Rossi et al., 2020). They can be divided into community-based and role-based (Rossi et al., 2020). Community-based embeddings are proximity-preserving and emphasize local connectivity patterns. This includes random walk-based methods that model co-occurrence statistics of nodes visited together on random walks (e.g., node2vec (Grover & Leskovec, 2016), LINE (Tang et al., 2015), ComE+ (Cavallari et al., 2019)), and feature propagation-based methods that diffuse the features through the neighborhoods using

GNNs (e.g. GCN (Kipf & Welling, 2017) and GraphSAGE (Hamilton et al., 2017)). While role-based embeddings are structural similarity and functional role-preserving, and focus on capturing higher-level structural patterns that are invariant to graph distance. This includes graphlet-based methods that encode local subgraph statistics (e.g., DeepGL (Rossi et al., 2017), MCN (Lee et al., 2018), HONE (Rossi et al., 2018)), feature-based walk approaches that represent sequences of feature values in walks (e.g., role2vec (Ahmed et al., 2018), node2bits (Jin et al., 2019a), SimSum (Liu et al., 2018)), and feature-based matrix factorization techniques that decompose a structure feature matrix to compute role representations (e.g., rolX (Henderson et al., 2012), HERO (Ahmed et al., 2017), EMBER (Jin et al., 2019b)).

B Theoretical Analysis

B.1 Time and Memory Complexity of GraphAlFA

Notation. Let $M = |\mathcal{E}|$ denote the number of edges in the graph. Let d_f and d_s denote the dimensionalities of the attribute-derived and structure-derived views, respectively, and let d_p denote the latent projection dimension used in the projector–gate fusion module. Let $\mathcal{I}$ denote the mini-batch set used for supervised contrastive learning, with mini-batch size $|\mathcal{I}|$. For the spectral objective, let K denote the number of frequency bands, m the number of Lanczos steps, and R the number of random probes used in stochastic spectral slicing.

B.1.1 Pre-processing.

(i) Attribute-derived view. The attribute-derived view is constructed by explicitly concatenating ego, 1-hop, and 2-hop feature contexts, $\mathbf{X}_f = [\mathbf{X}^{(0)} \parallel \mathbf{X}^{(1)} \parallel \mathbf{X}^{(2)}]$, where $\mathbf{X}^{(0)} = \mathbf{X}$, $\mathbf{X}^{(1)} = \hat{\mathbf{A}}\mathbf{X}$, and $\mathbf{X}^{(2)} = \hat{\mathbf{A}}^2\mathbf{X}$. This requires two sparse–dense matrix multiplications with the normalized adjacency matrix. Each multiplication costs $O(MF)$, yielding a total preprocessing time $T_{\text{attr-pre}} = O(M\,F)$, up to a constant factor of 2. The resulting attribute-derived feature matrix has dimensionality $d_f = 3F$, and memory

$$\text{Mem}(\mathbf{X}_f) = O(N\,d_f) = O(3NF).$$

The attribute-derived view is computed offline once prior to training and reused throughout training.

(ii) Structure-derived View. The structure-derived view $\mathbf{X}_s$ is computed once prior to training. Its cost depends on the chosen method. In our implementation, we adopt LINE (Tang et al., 2015) as a representative choice due to its scalability, with runtime linear in the number of edges. We denote the corresponding pre-processing cost as $T_{\text{str-pre}}$ and memory $\text{Mem}(\mathbf{X}_s) = O(Nd_s)$. The structure-derived view is computed offline once prior to training and reused throughout training.

B.1.2 Projector-Gate fusion.

The attribute- and structure-derived views are first projected into a shared latent space via $Z_f = \mathbf{X}_f W_{\text{attr}}$ with $W_{\text{attr}} \in \mathbb{R}^{d_f \times d_p}$ and $Z_s = \mathbf{X}_s W_{\text{str}}$ with $W_{\text{str}} \in \mathbb{R}^{d_s \times d_p}$, incurring

$$T_{Z_f} = O(N\,d_f\,d_p), \qquad T_{Z_s} = O(N\,d_s\,d_p).$$

The gate computes $G = \sigma(Z_f W_f + Z_s W_s)$ with $W_f, W_s \in \mathbb{R}^{d_p \times d_p}$, costing $T_{\text{gate}} = O(N\,d_p^2)$, plus $O(Nd_p)$ for element-wise operations. Finally, a linear residual mapping $r : \mathbb{R}^{d_p} \to \mathbb{R}^F$ incurs $T_{\text{res}} = O(N\,d_p\,F)$.

Overall, the forward cost of the projector-fusion module is:

$$T_{\text{fusion}} = O\big(N(d_f d_p + d_s d_p + d_p^2 + d_p F)\big),$$

with backward propagation of the same order.

Importantly, this cost is linear in the number of nodes and only involves linear transformations with small latent dimensions. It introduces no additional sparse graph operations. As a result, the projector-gate fusion has the same cost as a small number of standard fully connected layers and is dominated by the model message-passing cost. In practice, this module does not affect the overall scalability of the framework.

B.1.3 FEATURE-LABEL ALIGNMENT (CONTRASTIVE LOSS)

Applying the contrastive loss on a mini-batch of size $|\mathcal{I}|$ requires computing a $|\mathcal{I}| \times |\mathcal{I}|$ similarity matrix via dot products in $\mathbb{R}^{d_p}$,

$$T_{\text{SC}} = O(|\mathcal{I}|^2\, d_p), \qquad \text{Mem}_{\text{SC}} = O(|\mathcal{I}|^2).$$

Overall, the feature-label alignment objective introduces no graph-dependent overhead and scales well to large-scale graphs. Since its computational cost depends only on the batch size and the latent dimension, it is independent of the graph size.

B.1.4 FEATURE-STRUCTURE ALIGNMENT (SPECTRAL LOSS)

(i) Stochastic Lanczos spectral slicing. We analyze one computation that estimates the band-energy profile $E_S \in \mathbb{R}^K$ for a signal $S \in \mathbb{R}^{N\times d}$.

Lanczos tridiagonalization. Each of m Lanczos steps performs one sparse matrix-vector product $\hat{\mathcal{L}}q$ with cost $O(M)$ per step and per probe: $O(mM)$ per probe. Since m is a small constant in practice, this step scales linearly with the number of edges.

Tridiagonal eigen decomposition. Diagonalizing the resulting $m \times m$ tridiagonal matrix costs $O(m^3)$ per probe. As $m \ll N$ and is fixed, this cost is constant and negligible compared to sparse graph operations.

Band-pass projection in Krylov space. For fixed Lanczos basis $Q \in \mathbb{R}^{N\times m}$ and signals S,

$$\begin{aligned} Q^\top S &: \; O(N\, m\, d), \\ F_k(Q^\top S) &: \; O(m^2 d), \\ Q(\cdot) &: \; O(N\, m\, d). \end{aligned}$$

Across K bands yields $O\big(K(Nmd + m^2 d)\big)$ per probe. With fixed m and K, the dominant term scales linearly in the number of nodes.

Combining the above costs across R probes, the total time complexity of an energy-profile estimate is:

$$T_{\text{energy}}(d) = O\big(R\left(mM + m^3 + K(Nmd + m^2 d)\right)\big).$$

Under typical settings where m, R, and K are small constants, this procedure is linear in the graph size (N, M).

The corresponding working memory usage is

$$\begin{aligned} \text{Mem}_{\text{energy}}(d) = O(Nm) &\ (\text{store } Q) \\ +\ O(md) &\ (\text{store } Q^\top S) \\ +\ O(m^2) &\ (\text{store } T, U) \\ +\ O(Nd) &\ (\text{input } S). \end{aligned}$$

where band energies can be accumulated online to avoid storing intermediate band-pass projections. As a result, memory usage is linear in the number of nodes, with only small constant-factor overhead from the Krylov subspace.

(ii) Spectral Alignment loss. The spectral loss requires energy profiles for the fused features $\tilde{\mathbf{X}} \in \mathbb{R}^{N\times d_p}$ and the train-masked one-hot labels $\mathbf{Y}_{\text{tr}} \in \mathbb{R}^{N\times C}$. Thus, per epoch, this yields:

$$\begin{aligned} T_{\text{spec-per-epoch}} &= T_{\text{energy}}(d_p) + T_{\text{energy}}(C) \\ &= O\left(R\begin{pmatrix} 2mM + 2m^3 \\ +\, KNm(d_p{+}C) \\ +\, Km^2(d_p{+}C) \end{pmatrix}\right). \end{aligned}$$

Under typical settings where m, R, and K are fixed and small, the dominant terms scale linearly with the graph size (N, M) and the feature dimensions (d_p, C). The m^3 and m^2 terms are constant and therefore do not affect scalability. The cosine computation is $O(K)$.

(iii) Empirical filter estimation. Estimating the empirical propagation profile compares band energy distributions between consecutive layers $\mathbf{H}^{(\ell)} \to \mathbf{H}^{(\ell+1)}$. If performed every U epochs and across L layers, the update cost is:

$$T_{\text{gemp-update}} = \sum_{\ell=0}^{L-1} \left(T_{\text{energy}}(d_\ell) + T_{\text{energy}}(d_{\ell+1})\right) \approx O(2L \cdot T_{\text{energy}}(d_{\max}))\,,$$

where $d_{\max} = \max_\ell d_\ell$. Amortized over training, this contributes $\frac{1}{U}\, T_{\text{gemp-update}}$.

B.1.5 Overall runtime.

Combining the above components, the overall computational cost of GraphALFA scales favorably with graph size. All preprocessing steps are linear in graph size and are performed once offline prior to training. During training, the projector-gate fusion introduces only node-wise dense linear transformations with cost linear in N and small latent dimensions, and does not involve any graph-dependent operations. The feature-label contrastive objective depends only on the minibatch size and latent dimension, and is independent of the graph size. The spectral alignment objective, while graph-dependent, relies on stochastic Lanczos spectral slicing with small constant hyperparameters. Under typical settings, its dominant terms scale linearly with (N, M). Moreover, empirical filter estimation is performed periodically and amortized across epochs, contributing only a small additional overhead. As a result, GraphALFA only introduces modest computational overhead.

B.1.6 Peak memory footprint (training).

Persistent memory is dominated by:

$$O(M)\ \text{(sparse graph)}, \quad O(Nd_f)\ (\mathbf{X}_f), \quad O(Nd_s)\ (\mathbf{X}_s),$$

$$O(NF)\ (\mathbf{X}, \Delta), \quad O(Nd_p)\ (\tilde{\mathbf{X}}),$$

along with backbone activations $O(N\sum_{\ell=0}^{L} d_\ell)$ and optimizer state. Spectral slicing introduces a transient $O(Nm + md + m^2)$ buffer per probe, while contrastive loss adds a transient $O(|\mathcal{I}|^2)$ term.

B.2 Baseline Recoverability via Hypothesis Containment

The feature augmentation module intervenes only through the input mapping, preserving an explicit residual identity path to the original backbone input. Thus, it modifies the input representation without changing the backbone architecture or its parameterization. Let $\varphi \in \Phi$ denote the backbone model parameters, and $\theta \in \Theta$ the augmentation module parameters. We characterize the resulting baseline recoverability property of GraphALFA in terms of hypothesis containment.

Let $\mathbf{X} \in \mathbb{R}^{N\times F}$ be the node feature matrix, and let $\mathbf{X}_{\text{base}} \in \mathbb{R}^{N\times F}$ denote the feature matrix consumed by the backbone. The augmentation module injects a residual update in the same feature space through a row-wise readout $r_\theta : \mathbb{R}^{d_p} \to \mathbb{R}^F$:

$$\begin{aligned} \Delta\mathbf{X}_\theta(\mathbf{A}, \mathbf{X}) &:= r_\theta\big(\tilde{\mathbf{X}}_\theta(\mathbf{A}, \mathbf{X})\big) \in \mathbb{R}^{N\times F}, \\ \mathbf{X}_{\text{in}} &:= \mathbf{X}_{\text{base}} + \Delta\mathbf{X}_\theta(\mathbf{A}, \mathbf{X}). \end{aligned} \tag{2}$$

If the residual readout realizes the zero map, i.e., there exists $\theta_0 \in \Theta$ such that $\Delta\mathbf{X}_{\theta_0}(\mathbf{A}, \mathbf{X}) \equiv \mathbf{0}$, then $\mathbf{X}_{\text{in}} = \mathbf{X}_{\text{base}}$, exactly recovering the baseline input. Any global residual scaling can be absorbed into r_θ.

Lemma 2 (Baseline recoverability)**.** *Let*

$$\mathcal{H}_{\text{base}} := \big\{ f_\varphi(\mathbf{A}, \mathbf{X}_{\text{base}}) : \varphi \in \Phi \big\}$$

denote the hypothesis class induced by the baseline backbone, and let

$$\mathcal{H}_{\text{aug}} := \big\{ f_\varphi(\mathbf{A}, \mathbf{X}_{\text{base}} + \Delta\mathbf{X}_\theta(\mathbf{A}, \mathbf{X})) : \varphi \in \Phi, \theta \in \Theta \big\}$$

denote the hypothesis class induced by the augmented model. If the residual readout can realize the zero map, i.e., there exists $\theta_0 \in \Theta$ such that $\Delta\mathbf{X}_{\theta_0}(\mathbf{A}, \mathbf{X}) \equiv \mathbf{0}$, then

$$\mathcal{H}_{\text{base}} \subseteq \mathcal{H}_{\text{aug}}.$$

Consequently, for any empirical risk $\widehat{\mathcal{R}}_{\mathcal{V}_{\text{tr}}}$ on the labeled training nodes $\mathcal{V}_{\text{tr}}$,

$$\inf_{h \in \mathcal{H}_{\text{aug}}} \widehat{\mathcal{R}}_{\mathcal{V}_{\text{tr}}}(h) \leq \inf_{h \in \mathcal{H}_{\text{base}}} \widehat{\mathcal{R}}_{\mathcal{V}_{\text{tr}}}(h).$$

Proof. For any predictor in $\mathcal{H}_{\text{base}}$, there exists some $\varphi \in \Phi$ such that $h = f_\varphi(\mathbf{A}, \mathbf{X}_{\text{base}})$. Choose the same φ in the augmented model and set $\theta = \theta_0$, where $\Delta\mathbf{X}_{\theta_0}(\mathbf{A}, \mathbf{X}) \equiv \mathbf{0}$. Then equation 2 gives $\mathbf{X}_{\text{in}} = \mathbf{X}_{\text{base}}$, so the augmented predictor equals $f_\varphi(\mathbf{A}, \mathbf{X}_{\text{base}})$ point-wise. Thus every baseline predictor is contained in $\mathcal{H}_{\text{aug}}$, proving $\mathcal{H}_{\text{base}} \subseteq \mathcal{H}_{\text{aug}}$. □

Remark. Lemma 2 gives a hypothesis-class guarantee: the augmented model contains the baseline backbone as a special case and can exactly recover its input behavior when the residual update is zero. Thus, the augmentation is non-worsening in expressivity.

Since the augmentation enlarges the hypothesis class, it may also affect generalization; App. B.3 formalizes this trade-off through a transductive PAC-Bayes bound on the risk over the unlabeled nodes, where the module-specific term $\mathrm{KL}(Q_{\text{aug}} \| P_{\text{aug}})$ against a label-independent, baseline-centered prior can be interpreted as an augmentation complexity budget.

B.3 Augmentation Complexity Budget: A Transductive PAC-Bayes Generalization Lens

We adopt a transductive PAC-Bayes view (Bégin et al., 2014). Let V be the fixed node set with $N := |V|$, and let the labeled training subset $\mathcal{V}_{\text{tr}} \subseteq V$ be sampled uniformly without replacement from V, with $\mathcal{V}_{\text{te}} := V \setminus \mathcal{V}_{\text{tr}}$.

Loss and Gibbs risk. Assume a bounded loss $\ell \in [0, 1]$. For a distribution Q over predictors (or parameters), define the Gibbs risk on any node subset $\mathcal{V} \subseteq V$ as

$$\mathcal{R}_{\mathcal{V}}(Q) \;:=\; \mathbb{E}_{w \sim Q}\left[\frac{1}{|\mathcal{V}|}\sum_{i \in \mathcal{V}} \ell\big(h_w(i), y_i\big)\right].$$

Write $m_{\text{tr}} := |\mathcal{V}_{\text{tr}}|$ and note that the Gibbs risk over the *whole* node set decomposes as $\mathcal{R}_V(Q) = \frac{m_{\text{tr}}}{N}\mathcal{R}_{\mathcal{V}_{\text{tr}}}(Q) + \frac{N - m_{\text{tr}}}{N}\mathcal{R}_{\mathcal{V}_{\text{te}}}(Q)$.

For any prior P fixed before observing the training labels (i.e., independent of $\{y_i\}_{i \in \mathcal{V}_{\text{tr}}}$), with probability at least $1 - \delta$ over the random choice of $\mathcal{V}_{\text{tr}}$, a standard transductive PAC-Bayes inequality (Bégin et al., 2014) bounds the *full-population* risk:

$$\begin{aligned}\mathcal{R}_V(Q) \;\leq\; & \mathcal{R}_{\mathcal{V}_{\text{tr}}}(Q) \\ & + \sqrt{\Big(1 - \tfrac{m_{\text{tr}}}{N}\Big)\frac{\mathrm{KL}(Q\|P) + c(N, m_{\text{tr}}, \delta)}{2(m_{\text{tr}} - 1)}},\end{aligned} \tag{3}$$

where $c(N, m, \delta)$ is the explicit finite-population term (Bégin et al., 2014).

Solving the decomposition above for $\mathcal{R}_{\mathcal{V}_{\text{te}}}$ and substituting equation 3 gives the corresponding held-out statement, in which the factor $\sqrt{1 - m_{\text{tr}}/N}$ is replaced by $\sqrt{N/(N - m_{\text{tr}})}$.

No label leakage (prior admissibility). In the transductive setting, the prior P may depend on unlabeled information (e.g., $\mathbf{A}$ and $\mathbf{X}$), but must be fixed before observing the training labels (Bégin et al., 2014). Accordingly, a natural admissible choice is P to be *baseline-centered* at a label-independent reference point (e.g., baseline-compatible initialization).

Interpretation: augmentation complexity budget. The KL divergence $\mathrm{KL}(Q\|P)$ can be interpreted as a budget controlling how far the learned augmentation/posterior Q can deviate from the (label-independent) baseline-centered prior P; larger deviation increases the generalization penalty.

Factorized prior/posterior (backbone $\times$ augmentation). Let parameters split as $w = (w_{\text{base}}, w_{\text{augment}})$ (backbone vs. augmentation), and assume factorized measures $Q = Q_{\text{base}} \otimes Q_{\text{augment}}$ and $P = P_{\text{base}} \otimes P_{\text{augment}}$. Then $\mathrm{KL}(Q\|P) =$

$$\mathrm{KL}(Q_{\text{base}}\|P_{\text{base}}) + \mathrm{KL}(Q_{\text{augment}}\|P_{\text{augment}}). \tag{4}$$

In our setting, we take P_{augment} baseline-centered, so $\mathrm{KL}(Q_{\text{augment}}\|P_{\text{augment}})$ isolates augmentation complexity.

Closed form (Gaussian instantiation). For isotropic Gaussians over $\mathbb{R}^{d_{\text{augment}}}$: $Q_{\text{augment}} = \mathcal{N}(\hat{w}_{\text{augment}}, \sigma^2 I)$ and $P_{\text{augment}} = \mathcal{N}(w_{0,\text{augment}}, \sigma_0^2 I)$. Then $\mathrm{KL}(Q_{\text{augment}}\|P_{\text{augment}}) =$

$$\frac{1}{2}\left(\frac{\|\hat{w}_{\text{augment}} - w_{0,\text{augment}}\|_2^2}{\sigma_0^2} + d_{\text{augment}}\left(r - 1 - \log r\right)\right), \tag{5}$$

where $r := \sigma^2/\sigma_0^2$. Thus staying close to a baseline-centered prior reduces the complexity term in equation 3.

Optimization-friendly surrogate. Using $\sqrt{u} \le \frac{u}{2\beta} + \frac{\beta}{2}$ for any $\beta > 0$ and $u = \left(1 - \frac{m_{\text{tr}}}{N}\right)\frac{\mathrm{KL}(Q\|P)+c(N,m_{\text{tr}},\delta)}{2(m_{\text{tr}}-1)}$, equation 3 yields an upper bound of the form

$$\mathcal{R}_V(Q) \le \mathcal{R}_{\mathcal{V}_{\text{tr}}}(Q) + \left(1 - \tfrac{m_{\text{tr}}}{N}\right)\frac{\mathrm{KL}(Q\|P)}{4\beta\,(m_{\text{tr}} - 1)} + (\text{constants}).$$

This motivates training objectives that explicitly regularize toward the baseline (e.g., $\widehat{\mathcal{R}}_{\mathcal{V}_{\text{tr}}} + \lambda\,\mathrm{KL}(Q\|P)$ with $\lambda = \left(1 - \frac{m_{\text{tr}}}{N}\right)\left(4\beta(m_{\text{tr}} - 1)\right)^{-1}$, which is realized through the output-space regularizer $\lambda_{\text{reg}}\|\Delta\mathbf{X}\theta\|_F^2$, which likewise discourages large interventions and favors gains achieved with minimal deviation from $\mathbf{X}\text{base}$.

Remark (Gibbs vs. deterministic predictor). The PAC-Bayes bound applies to the Gibbs predictor induced by Q; we use it as a complexity-motivated design lens rather than claiming the same expression directly bounds the deterministic predictor.

B.4 Feature-Label Information Coupling via Contrastive Learning (Sec. 4.1)

We justify the directional chain used in Sec. 4.1:

$$I(Z;Y) \ge I(Z;Z^+) \gtrsim \log(I-1) - \mathbb{E}\big[\mathcal{L}_{\text{SC}}(\mathcal{I})\big]. \tag{6}$$

Here Z denotes an anchor embedding (a row of $\tilde{\mathbf{X}}$), Z^+ denotes a *positive* embedding (another node embedding with the same training label), Y denotes the training-label random variable, and I is the mini-batch size. All information quantities and loss terms are interpreted at the population level; $\mathbb{E}[\cdot]$ is over the data distribution and the mini-batch/positive/negative sampling used to form $\mathcal{I}$. Crucially, positives are constructed using labels only from $\mathcal{V}_{\text{tr}}$; no labels from $\mathcal{V}_{\text{te}}$ are used. Here $\gtrsim$ indicates the bound holds under the same sampling/critic assumptions as InfoNCE and after the standard SupCon-to-per-pair relaxation described below.

Step 1: DPI under a class-conditional abstraction. Following standard supervised-contrastive analyses (Khosla et al., 2020), we adopt the population-level abstraction that

$$Z \sim p_\theta(z \mid Y), \qquad Z^+ \sim p_\theta(z \mid Y), \qquad Z \perp\!\!\!\perp Z^+ \mid Y,$$

i.e., $Z \leftarrow Y \rightarrow Z^+$ is a Markov structure (a common-cause model). Then the Data Processing Inequality implies $I(Z;Z^+) \le I(Z;Y)$, which yields the left inequality in equation 6 (Cover & Thomas, 2006).

Step 2: InfoNCE lower bound for a $(I-1)$-candidate softmax. Let $\mathcal{I}$ be a mini-batch and $I := |\mathcal{I}|$. Consider an InfoNCE loss with one designated positive and $I-2$ negatives:

$$\mathcal{L}_{\mathrm{NCE}} := -\mathbb{E}\left[\log \frac{f(Z, Z^+)}{f(Z, Z^+) + \sum_{j=1}^{I-2} f(Z, Z_j^-)}\right],$$

where f is a nonnegative critic and (standard abstraction) negatives Z_j^- are treated as draws from the marginal. Under the standard assumptions used in InfoNCE analyses—(i) $Z_j^- \overset{iid}{\sim} p_\theta(z)$ and independent of Z; and (ii) the critic family is sufficiently rich to represent (or approximate) the optimal density-ratio critic and is optimized to (near-)optimality—one obtains the variational bound (van den Oord et al., 2018; Poole et al., 2019)

$$I(Z; Z^+) \;\geq\; \log(I-1) - \mathcal{L}_{\mathrm{NCE}}. \tag{7}$$

From multi-positive SupCon to InfoNCE-style terms. SupCon (Khosla et al., 2020) averages anchor–positive log-softmax terms. For an anchor $\tilde{\mathbf{x}}_i$ and its positive set $\mathcal{P}(i)$,

$$\mathcal{L}_{\mathrm{SC}}(i) = -\frac{1}{|\mathcal{P}(i)|} \sum_{j \in \mathcal{P}(i)} \log \frac{\exp\big(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{x}}_j / \tau\big)}{\sum_{k \in \mathcal{I} \setminus \{i\}} \exp\big(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{x}}_k / \tau\big)}.$$

For each fixed anchor–positive pair (i, j), the denominator ranges over the $I-1$ non-anchor samples, matching the candidate-size convention in equation 7. Applying equation 7 at the per-pair level and then averaging over positives and anchors yields the commonly used directional inequality

$$I(Z; Z^+) \;\gtrsim\; \log(I-1) - \mathbb{E}\big[\mathcal{L}_{\mathrm{SC}}(\mathcal{I})\big],$$

up to the same sampling/critic/optimization assumptions used in the InfoNCE analyses.

Remark (graph dependence and sampling). On graphs, samples within a batch may be correlated (e.g., due to homophily or structural proximity), so the "negatives from the marginal" abstraction can be imperfect. We therefore use equation 6 as a population-level, directional justification for why minimizing $\mathcal{L}_{\mathrm{SC}}$ encourages label-informative embeddings, rather than as an exact finite-sample identity guarantee. □

B.5 Propagation-Aware Spectral Matching for Feature-Structure Alignment (Sec. 4.2)

Proposition 2 (Cosine loss equals normalized-profile mismatch)**.** *Assume* $\mathbf{w} \geq \mathbf{0}$ *with* $\sqrt{\mathbf{w}} \odot \mathbf{E}_f \neq \mathbf{0}$ *and* $\sqrt{\mathbf{w}} \odot \mathbf{E}_y \neq \mathbf{0}$*, and define* $\mathbf{a} := \sqrt{\mathbf{w}} \odot \mathbf{E}_f$*,* $\mathbf{b} := \sqrt{\mathbf{w}} \odot \mathbf{E}_y$*. Then*

$$\left\| \frac{\mathbf{a}}{\|\mathbf{a}\|_2} - \frac{\mathbf{b}}{\|\mathbf{b}\|_2} \right\|_2^2 = 2\left(1 - \frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\|\mathbf{a}\|_2 \, \|\mathbf{b}\|_2}\right) = 2\,\mathcal{L}_{\mathrm{spec}}.$$

Moreover $\mathcal{L}\mathrm{spec} \in [0, 1]$*, and* $\mathcal{L}\mathrm{spec} = 0$ *exactly when* $\mathbf{E}_f$ *and* $\mathbf{E}_y$ *agree up to a positive scale factor on the bands with* $w_k > 0$*.*

Proof. Since $\mathbf{a}/|\mathbf{a}|_2$ and $\mathbf{b}/|\mathbf{b}|_2$ are unit vectors, expanding the square gives $2 - 2\cos(\mathbf{a}, \mathbf{b})$. With $\mathbf{a} = \sqrt{\mathbf{w}} \odot \mathbf{E}_f$ and $\mathbf{b} = \sqrt{\mathbf{w}} \odot \mathbf{E}_y$ we have $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_k w_k E_{f,k} E_{y,k}$, $\|\mathbf{a}\|_2^2 = \sum_k w_k E_{f,k}^2$ and $\|\mathbf{b}\|_2^2 = \sum_k w_k E_{y,k}^2$, so $\cos(\mathbf{a}, \mathbf{b})$ is exactly the weighted cosine of equation 1 and the right-hand side equals $2\mathcal{L}_{\mathrm{spec}}$. Since $\mathbf{w}, \mathbf{E}_f, \mathbf{E}_y \geq \mathbf{0}$, the vectors $\mathbf{a}, \mathbf{b}$ are entrywise nonnegative, so $\langle \mathbf{a}, \mathbf{b} \rangle \geq 0$ and $\cos(\mathbf{a}, \mathbf{b}) \in [0, 1]$, with equality to 1 iff Cauchy–Schwarz is tight, i.e. $\mathbf{a} \propto \mathbf{b}$ — a condition on the coordinates with $w_k > 0$ only. □

Lemma 3 (Bandwise energy bound)**.** *Let* $\mathbf{T} = \mathbf{U}\,\mathrm{diag}(t(\lambda_0), \ldots, t(\lambda_{N-1}))\mathbf{U}^\top$*.*

For each band $\mathcal{B}_k$*, define the discrete extrema over eigenvalues in the band:* $t_{\min,k} := \min_{i:\lambda_i \in \mathcal{B}_k} |t(\lambda_i)|$ *and* $t_{\max,k} := \max_{i:\lambda_i \in \mathcal{B}_k} |t(\lambda_i)|$*. Then for any* $\mathbf{S} \in \mathbb{R}^{N \times d}$ *and any* $k \in [K]$*,*

$$t_{\min,k}^2 \, E_{\mathbf{S}}(k) \;\leq\; E_{\mathbf{TS}}(k) \;\leq\; t_{\max,k}^2 \, E_{\mathbf{S}}(k).$$

Proof. Let $\widehat{\mathbf{S}} := \mathbf{U}^\top \mathbf{S} \in \mathbb{R}^{N\times d}$. Then

$$E_{\mathbf{S}}(k) = \|\mathbf{\Pi}_k \mathbf{S}\|_F^2 = \|\mathbf{U}\mathbf{M}_k \widehat{\mathbf{S}}\|_F^2 = \|\mathbf{M}_k \widehat{\mathbf{S}}\|_F^2.$$

$$\begin{aligned} E_{\mathbf{TS}}(k) = \|\mathbf{\Pi}_k \mathbf{T}\mathbf{S}\|_F^2 &= \|\mathbf{U}\mathbf{M}_k \operatorname{diag}(t)\widehat{\mathbf{S}}\|_F^2 \\ &= \|\mathbf{M}_k \operatorname{diag}(t)\widehat{\mathbf{S}}\|_F^2. \end{aligned}$$

Writing row i of $\widehat{\mathbf{S}}$ as $\widehat{\mathbf{s}}_i^\top \in \mathbb{R}^{1\times d}$,

$$\|\mathbf{M}_k \operatorname{diag}(t)\widehat{\mathbf{S}}\|_F^2 = \sum_{i:\lambda_i\in\mathcal{B}_k} |t(\lambda_i)|^2 \, \|\widehat{\mathbf{s}}_i\|_2^2.$$

Since $|t(\lambda_i)|^2 \in [t_{\min,k}^2, t_{\max,k}^2]$ for all i in the band,

$$\begin{aligned} t_{\min,k}^2 \sum_{i:\lambda_i\in\mathcal{B}_k} \|\widehat{\mathbf{s}}_i\|_2^2 \le \sum_{i:\lambda_i\in\mathcal{B}_k} |t(\lambda_i)|^2 \, \|\widehat{\mathbf{s}}_i\|_2^2 \\ \le t_{\max,k}^2 \sum_{i:\lambda_i\in\mathcal{B}_k} \|\widehat{\mathbf{s}}_i\|_2^2, \end{aligned}$$

and the outer sums are exactly $E_{\mathbf{S}}(k)$. □

Connecting $\mathbf{g}_{\text{emp}}$ to transmission. A convenient *label-free* empirical estimate of bandwise transmission is the band energy gain ratio for a probe signal $\mathbf{R} \in \mathbb{R}^{N\times d}$:

$$g_{\text{emp},k} \;:=\; \frac{E_{\mathbf{TR}}(k)}{E_{\mathbf{R}}(k)} \qquad \text{(defined when } E_{\mathbf{R}}(k) > 0\text{)}.$$

By Lemma 3, whenever $E_{\mathbf{R}}(k) > 0$ we have

$$t_{\min,k}^2 \;\le\; g_{\text{emp},k} \;\le\; t_{\max,k}^2.$$

In practice one may average this ratio across layers and/or multiple label-free probes to stabilize the estimate, yielding the $\mathbf{g}_{\text{emp}}$ used to form weights $\mathbf{w} \propto$ ‘$\mathbf{g}_{\text{emp}}$ in $\mathcal{L}_{\text{spec}}$.

C PSEUDOCODE

The algorithms below summarize the main computational components of GRAPHALFA.

Algorithm 1 End-to-end Training

Input: Graph $\mathcal{G}$, sparse normalized adjacency $\hat{\mathbf{A}}$, Laplacian matvec oracle $\hat{\mathcal{L}}(\cdot)$, features $\mathbf{X} \in \mathbb{R}^{N\times F}$, training labels $\{y_i\}_{i\in\mathcal{V}_{\text{tr}}}$ (one-hot $\mathbf{Y}_{\text{tr}}$), backbone f_φ; spectral bands $\{\mathcal{B}_k\}_{k=1}^{K}$; Lanczos params (R, m); epochs $T_{\max}$; warmup W; filter-update interval U; SupCon mini-batch size I; loss weights $(\lambda_{\text{SC}}, \lambda_{\text{spec}}, \lambda_{\text{reg}})$.
Output: Trained augmentation module parameters θ and backbone parameters φ.

/* Attribute-derived view: fixed 2-hop context */
$\mathbf{X}^{(0)} \leftarrow \mathbf{X}$
$\mathbf{X}^{(1)} \leftarrow \hat{\mathbf{A}}\,\mathbf{X}^{(0)}$ ▷ sparse-dense multiply
$\mathbf{X}^{(2)} \leftarrow \hat{\mathbf{A}}\,\mathbf{X}^{(1)}$ ▷ sparse-dense multiply
$\mathbf{X}_f \leftarrow [\mathbf{X}^{(0)} \| \mathbf{X}^{(1)} \| \mathbf{X}^{(2)}]$
/* Structure-derived view (pre-computed) */
$\mathbf{X}_s \leftarrow \textsc{StructEmbed}(\mathcal{G})$ ▷ offline and fixed, method-dependent
Initialize augmentation module parameters $\theta = (W_{\text{attr}}, W_{\text{str}}, W_f, W_s, r)$ and backbone parameters φ
$g_{\text{emp}} \leftarrow \mathbf{1} \in \mathbb{R}^K$
for $t = 1$ to $T_{\max}$ **do**
  if $t > W$ **and** $t \bmod U = 0$ **then** ▷ empirical filter update
    $(\mathbf{X}_{\text{in}}, \tilde{\mathbf{X}}, \Delta) \leftarrow \textsc{AugmentedForward}(\mathbf{X}, \mathbf{X}_f, \mathbf{X}_s; \theta)$
    $(\cdot, \mathbf{H}^{(0)}, \ldots, \mathbf{H}^{(L)}) \leftarrow \textsc{BackboneWithActs}(f_\varphi, \hat{\mathbf{A}}, \mathbf{X}_{\text{in}})$ ▷ no-grad
    for $\ell = 0$ to $L-1$ **do**
      $E_\ell \leftarrow \textsc{BandEnergy}(\mathbf{H}^{(\ell)})$
      $E_{\ell+1} \leftarrow \textsc{BandEnergy}(\mathbf{H}^{(\ell+1)})$
      $\text{gain}_\ell \leftarrow E_{\ell+1} \oslash (E_\ell + \varepsilon)$
    end for
    $g_{\text{emp}} \leftarrow \textsc{Mean}(\{\text{gain}_\ell\}_{\ell=0}^{L-1})$ ▷ element-wise over the K bands
  end if
  $w \leftarrow \textsc{Normalize}(\max(g_{\text{emp}}, 0))$
  $(\mathbf{X}_{\text{in}}, \tilde{\mathbf{X}}, \Delta) \leftarrow \textsc{AugmentedForward}(\mathbf{X}, \mathbf{X}_f, \mathbf{X}_s; \theta)$
  $\hat{y} \leftarrow f_\varphi(\hat{\mathbf{A}}, \mathbf{X}_{\text{in}})$
  $L_{\text{task}} \leftarrow \textsc{TaskLoss}(\hat{y}[\mathcal{V}_{\text{tr}}],\, y[\mathcal{V}_{\text{tr}}])$
  $L_{\text{reg}} \leftarrow \|\Delta\|_F^2$
  Sample minibatch $\mathcal{I} \subseteq \mathcal{V}_{\text{tr}}$ with $|\mathcal{I}| = I$
  $L_{\text{SC}} \leftarrow \textsc{SupCon}(\tilde{\mathbf{X}}[\mathcal{I}, :], y[\mathcal{I}])$
  $E_{\tilde{\mathbf{X}}} \leftarrow \textsc{BandEnergy}(\tilde{\mathbf{X}})$
  $E_{\mathbf{Y}_{\text{tr}}} \leftarrow \textsc{BandEnergy}(\mathbf{Y}_{\text{tr}})$
  $\mathcal{L}_{\text{spec}} \leftarrow 1 - \textsc{WeightedCosine}(E_{\tilde{\mathbf{X}}}, E_{\mathbf{Y}_{\text{tr}}}, w)$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{task}} + \lambda_{\text{SC}}\mathcal{L}_{\text{SC}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}$
  Update (θ, φ) by one gradient step on $\mathcal{L}$
end for
return (θ, φ)

D Experiments

D.1 Real-World Datasets

D.1.1 Setup.

For evaluation on real-world datasets, we consider a diverse set of benchmark datasets spanning a diverse range of domains, including Chameleon (Platonov et al., 2023), Squirrel (Platonov et al., 2023), Flickr (Zhou et al., 2023), Ratings (Platonov et al., 2023), Arxiv-Year (Lim et al., 2021), Cora (Yang et al., 2016), PubMed (Yang et al., 2016), WikiCS (Mernyei & Cangea, 2020), Citeseer (Yang et al., 2016), Ele-fashion (Zhu et al., 2025), and Goodreads-NC (Zhu et al., 2025). The

Algorithm 2 BANDENERGY(S) via stochastic Lanczos spectral slicing

Input: signal $S \in \mathbb{R}^{N\times d}$, Laplacian matvec oracle $\hat{\mathcal{L}}(\cdot)$, bands $\{\mathcal{B}_k\}_{k=1}^{K}$, probes R, Lanczos steps m.
Output: band-energy vector $E \in \mathbb{R}^K$ approximating $\|\Pi_k S\|_F^2$.

$E \leftarrow \mathbf{0} \in \mathbb{R}^K$
for $r = 1$ to R **do**
  Sample random $z \in \mathbb{R}^N$ and normalize
  Run m-step Lanczos on $\hat{\mathcal{L}}$ with start z to get $Q \in \mathbb{R}^{N\times m}$ and tridiagonal $T \in \mathbb{R}^{m\times m}$
  Eigendecompose $T = U\,\mathrm{diag}(\vartheta)U^\top$
  $S_{\text{kry}} \leftarrow Q^\top S$ ▷ size $m \times d$
  for $k = 1$ to K **do**
    Build diagonal mask $M_k(\vartheta)$ selecting $\vartheta \in \mathcal{B}_k$
    $F_k \leftarrow U M_k U^\top$ ▷ $m \times m$ band operator in Krylov space
    $\widehat{\Pi_k S} \leftarrow Q\,(F_k\, S_{\text{kry}})$ ▷ approx band-projection, size $N \times d$
    $E[k] \leftarrow E[k] + \|\widehat{\Pi_k S}\|_F^2$
  end for
end for
return E/R

Table 4: Summary statistics of real-world datasets used in our experiments.

Dataset	Type	#Nodes	#Edges	Global Edge Homophily
Chameleon	Web	890	8,854	0.2361
Squirrel	Web	2,223	46,998	0.2072
Flickr	Social	7,575	239,738	0.2386
Ratings	Co-Purchase	24,492	93,050	0.3804
Arxiv-year	Citation	169,343	1,166,243	0.2218
Cora	Citation	2,708	5,429	0.8100
PubMed	Citation	19,717	44,338	0.8024
WikiCS	Web	11,701	216,123	0.6547
Citeseer	Citation	3,327	4,732	0.7355
Ele-fashion	Co-Purchase	97,766	199,602	0.7675
Goodreads-NC	Books	685,294	7,235,084	0.6667

splits for training/validation/test are 60%/20%/20% random splits, following the setup from (Chien et al., 2021). For the Chameleon and Squirrel datasets, we adopt the filtered versions proposed in prior benchmarks (Platonov et al., 2023), which remove overlapping nodes between training and test sets.

Notably, some datasets provide rich node attributes beyond shallow features. Specifically, Ele-fashion and Goodreads-NC are multi-modal and include both text and image-based attributes, while Ratings, Arxiv-Year, Cora, PubMed, and WikiCS include textual attributes, enabling evaluation on multi-modal features from pretrained language or vision encoders.

For text-attributed graphs, textual attributes were encoded using the pretrained sentence-transformer model all-mpnet-base-v2 from the SENTENCE-TRANSFORMERS library. This setup, referred to as *LLM-as-enhancer* (embedding-based), has been shown to achieve improved performance over shallow features by producing semantically rich node embeddings. For multi-modal image-attributed graphs, we adopt the datasets from the MM-Graph (Zhu et al., 2025) benchmark, where each node is associated with both textual and image attributes (e.g. fashion product titles and images). Following the benchmark configuration, we use CLIP embeddings, which jointly align textual and image features within a unified embedding space and demonstrated the best performance.

Table 5: Node classification accuracy (%) on real-world datasets: Mean accuracy across runs ± standard deviation. We compare different graph feature augmentation methods on the heterophily-aware ACM-GCN model. Best average accuracy in bold.

Dataset	Original	Augmentation					
		Eigen-GNN	NetInfoF	GERF	LLPE	LAGNN	Ours
ACM-GCN							
Chameleon	42.14±1.3	45.42±3.4	47.12±5.6	44.29±4.0	44.44±3.1	44.26±4.5	**51.41±1.6**
Squirrel	46.66±5.6	48.34±5.0	42.68±5.9	47.76±2.1	46.96±0.6	39.41±5.0	**53.17±1.0**
Flickr	91.76±1.3	92.63±0.4	90.71±0.3	65.04±1.1	92.81±0.1	**93.75±0.3**	92.19±0.3
Ratings	51.98±2.2	52.49±0.3	50.68±0.8	48.92±0.5	48.17±3.0	36.67±7.1	**52.62±0.6**
Arxiv-Year	42.11±1.0	41.97±1.8	**49.79±0.1**	45.11±0.7	47.22±0.5	28.43±6.6	47.92±1.1
Cora	86.31±2.5	87.36±1.5	85.98±1.9	83.97±1.0	87.74±1.3	88.62±1.9	**88.62±1.8**
PubMed	88.93±1.4	89.32±0.7	87.91±0.4	84.43±0.4	88.83±0.7	**89.61±0.3**	89.24±0.6
WikiCS	83.90±0.5	83.73±0.8	**84.38±0.6**	82.56±0.5	83.33±0.1	77.17±1.1	83.40±0.4
Citeseer	78.90±0.5	78.82±1.4	77.61±0.6	72.66±1.1	79.85±1.2	80.08±1.1	**80.40±0.8**
Avg. Rank		2.9	3.6	5.0	3.6	3.9	**2.0**

D.1.2 RESULTS.

We further evaluate GRAPHALFA under more powerful feature representations, on a subset of real-world datasets with text or multimodal attributes. Specifically, for text-attributed graphs (Ratings, Arxiv-Year, Cora, PubMed, and WikiCS), the features are obtained by encoding textual attributes using a Pretrained Language Model (PLM). And for multi-modal graphs (Ele-fashion and Goodreads), the text and visual attributes are jointly encoded using the CLIP multimodal encoder. The multi-modal dataset results are presented in Table 6.

Across all backbone architectures, GRAPHALFA consistently maintains the best overall performance under these multi-modal feature representations, generalizing well across node representation modalities in multi-modal graph learning scenarios. Notably, these multi-modal feature encodings are generally more powerful than the original shallow features, and have been shown to be able to improve model performance. The performance improvements by the GRAPHALFA framework demonstrate that the proposed alignment-guided augmentation is robust to the choice of feature modality. This indicates that even with already powerful embeddings, misalignment in the graph data persists, and can be effectively mitigated and refined by our proposed augmentation framework.

D.2 SYNTHETIC DATASETS

D.2.1 SETUP.

Following the setup described in (Linkerhägner et al., 2025), the cSBM extends the classic Stochastic Block Model (Abbe, 2017) by assigning each node with high-dimensional features. Specifically, we generate graphs with $N = 2000$ nodes and fix the expected average degree to $d = 5$. To generate a balanced two-class graph with N nodes, each node is assigned a label $y_i \in \{\pm 1\}$, and pairs of nodes connect independently at random, such that edges between intra-class and inter-class nodes are sampled with probabilities $p_{in} = \frac{c_{in}}{N}$ and $p_{out} = \frac{c_{out}}{N}$, where $c_{in} = d + \sqrt{d}\Lambda$ and $c_{out} = d - \sqrt{d}\Lambda$. d denotes the expected average degree and Λ is a parameter representing the structural signal-to-noise ratio. Each node's p-dimensional feature $x_i \in \mathbb{R}^p$ is drawn from a Gaussian Mixture Model (GMM) $x_i = \sqrt{\frac{\mu}{N}} y_i \xi + \frac{z_i}{\sqrt{p}}$, where $\xi \sim \mathcal{N}(0, \frac{I_p}{p})$ is the randomly drawn mean, $z_i \sim \mathcal{N}(0, I_p)$ is the i.i.d noise, and μ regulates the strength of feature-label correlation. Following (Chien et al., 2021) and (Linkerhägner et al., 2025), we control the level of feature and structure signal using Λ and μ, and parameterize them in terms of two interpretable parameters ϵ and ϕ: $\Lambda = \sqrt{1+\epsilon} * sin(\frac{\pi}{2}\phi)$, $\mu = \sqrt{\gamma(1+\epsilon)} * cos(\frac{\pi}{2}\phi)$, where $\gamma = \frac{N}{p}$. ϵ controls the overall signal strength across structure and features, while ϕ controls the homophily level (balance between structure signal and feature signal strength). Throughout our experiments, we use $\epsilon = 2$ to ensure the presence of meaningful recoverable signal, and vary $\phi \in [-1, 1]$ using a uniform grid with step size 0.2.

Table 6: Node classification accuracy (%) using Pretrained Language Model or joint multi-modal encoder features. We evaluate different graph augmentation methods across various backbone models on datasets with textual or multi-modal attributes. Results are reported as mean accuracy across runs ± standard deviation. Best average accuracy in bold.

Dataset	Original	Augmentation					
		Eigen-GNN	NetInfoF	GERF	LLPE	LAGNN	Ours
MLP							
Ratings	52.99±0.3	52.23±0.9	52.67±0.9	50.40±0.7	46.86±0.5	37.36±0.5	**56.37±0.3**
Arxiv-Year	44.21±0.1	44.84±0.3	**51.33±0.3**	43.17±0.1	48.39±0.5	28.80±0.1	51.22±0.2
Cora	81.22±1.0	84.20±1.2	85.89±0.5	84.27±0.9	79.99±2.2	81.84±1.7	**87.66±0.4**
PubMed	90.97±0.4	90.70±0.3	89.96±0.3	85.64±0.4	89.43±0.4	**90.72±0.3**	90.09±0.2
WikiCS	83.64±0.9	80.48±0.8	86.06±0.4	84.07±0.7	83.21±0.5	71.41±2.1	**87.57±0.6**
Fashion	86.99±0.3	87.03±0.2	85.00±0.1	79.29±0.2	**87.61±0.1**	45.10±4.1	87.53±0.2
Goodreads	77.00±0.2	77.54±0.1	87.47±0.1	OOM	81.84±0.1	OOM	**88.84±0.1**
Avg. Rank		3.6	2.4	4.4	3.9	5.0	**1.6**
GCN							
Ratings	51.60±3.3	**54.28±0.9**	46.74±0.9	49.70±0.3	43.40±1.5	48.68±1.0	52.98±0.7
Arxiv-Year	50.26±0.2	47.75±0.8	48.69±0.7	49.27±0.2	50.96±0.1	43.24±1.0	**53.57±0.1**
Cora	88.09±0.7	88.15±1.7	87.66±1.0	86.70±1.2	88.16±1.1	84.94±0.8	**89.38±1.3**
PubMed	89.24±0.4	89.45±0.5	87.72±0.4	84.48±0.5	88.46±0.2	**90.99±0.8**	89.99±0.7
WikiCS	84.82±0.2	85.06±0.5	84.02±0.3	83.93±0.6	84.23±0.8	83.90±0.7	**86.19±0.6**
Fashion	84.35±0.2	85.13±0.4	83.86±0.3	81.06±0.3	86.44±0.4	85.45±0.2	**86.86±0.2**
Goodreads	84.24±0.2	84.77±0.3	**87.67±0.2**	OOM	84.78±0.2	OOM	87.57±0.1
Avg. Rank		3.0	4.0	4.7	3.1	4.4	**1.4**
GAT							
Ratings	50.71±0.7	51.41±0.4	51.02±0.6	53.19±0.8	39.23±2.8	46.51±2.2	**54.26±0.6**
Arxiv-Year	46.78±0.7	51.36±0.9	52.20±0.4	47.93±0.4	52.50±0.5	41.06±2.6	**53.48±0.1**
Cora	87.85±0.7	88.29±1.8	86.85±2.6	85.27±1.1	87.37±1.7	85.41±1.3	**89.38±0.7**
PubMed	88.43±0.4	89.84±0.4	89.21±0.4	83.11±0.8	89.14±0.2	**90.87±0.3**	90.00±0.3
WikiCS	84.07±0.6	86.63±0.4	85.48±0.4	82.67±0.7	84.17±0.5	83.10±1.0	**87.13±0.4**
Fashion	84.84±0.5	86.02±0.3	84.67±0.3	76.89±1.1	**86.75±0.2**	85.41±0.6	86.68±0.2
Goodreads	84.46±0.2	84.35±0.2	83.00±0.1	OOM	81.93±0.6	OOM	**87.11±0.1**
Avg. Rank		2.7	3.7	5.1	3.6	4.4	**1.3**
ACM-GCN							
Ratings	49.21±0.5	48.62±1.4	48.21±0.8	48.63±0.7	47.23±1.6	33.04±1.9	**49.62±0.7**
Arxiv-Year	48.25±1.1	48.37±0.4	**52.52±0.3**	46.42±0.4	51.05±0.2	23.26±1.9	49.46±1.1
Cora	89.20±1.1	88.21±0.7	86.60±1.6	84.11±1.0	89.11±0.9	88.29±1.5	**89.93±0.1**
PubMed	90.56±0.4	90.42±0.4	88.97±0.5	84.14±0.7	89.33±0.1	**90.57±0.4**	90.46±0.2
WikiCS	86.26±0.6	**86.25±0.3**	85.06±0.8	84.03±0.2	85.89±0.4	79.68±0.1	85.18±0.3
Fashion	89.15±0.3	**89.78±0.6**	85.81±0.2	68.83±0.2	89.64±1.1	83.51±1.6	88.44±0.9
Goodreads	79.58±1.1	79.81±0.1	**82.63±0.1**	OOM	79.66±1.0	OOM	78.47±0.6
Avg. Rank		2.6	3.4	5.0	2.9	4.6	**2.4**

D.2.2 RESULTS.

We also evaluate GRAPHALFA on cSBM synthetic datasets using the ACM-GCN model. Figure 3 reports accuracy comparison with baseline methods across varying homophily levels. Consistent with expectations, ACM-GCN substantially improves over standard GCN in heterophilic datasets, reflecting its explicit design for handling heterophily. However, at large $|\phi|$ values (extreme heterophily and extreme homophily), GRAPHALFA yields additional gains over the ACM-GCN baseline. This suggests that even when the backbone model is tailored to heterophily, our alignment-driven feature augmentation approach can provide complementary benefits.

D.3 ABLATION AND SENSITIVITY ANALYSIS

D.3.1 STRUCTURE-DERIVED VIEW CHOICES.

Table 7 compares different choices of structure-derived view on additional backbone models. Performance remains stable across embedding choices, this is consistent with the trends and observations we make on the GCN model in Section 5.2.

Table 7: Ablation study on structure-derived view choices within our proposed augmentation framework on real-world datasets. Mean accuracy across runs ± standard deviation.

Dataset	Node2Vec (Local)	Node2Vec (Global)	xNetMF	Role2vec	LINE	Top LPE	Top+ Bottom LPE
MLP							
Chameleon	44.76±1.7	43.63±1.5	47.57±1.7	48.31±1.2	48.13±1.3	49.06±1.9	47.75±2.8
Squirrel	44.77±1.5	44.99±2.6	43.12±0.6	43.27±2.1	43.57±2.3	42.38±3.7	42.08±0.9
Flickr	94.04±0.7	93.97±0.7	94.46±0.3	93.86±0.5	94.04±0.6	93.60±0.6	93.95±0.1
Ratings	50.00±0.6	50.30±0.4	55.46±0.3	58.07±0.9	57.38±0.3	51.60±0.1	52.08±0.1
Arxiv-Year	44.87±0.2	49.22±0.2	44.73±0.3	47.45±0.3	49.43±0.4	50.29±0.3	49.01±0.1
Cora	89.44±1.7	88.77±1.4	88.95±0.4	89.81±1.2	89.20±0.9	89.13±0.2	89.81±0.9
PubMed	89.00±0.2	89.66±0.3	89.60±0.1	89.65±0.2	89.00±0.3	89.94±0.2	89.82±0.4
WikiCS	86.06±0.3	86.76±0.5	85.18±0.6	83.91±0.4	85.89±0.5	86.33±0.2	85.62±0.3
Citeseer	78.89±1.1	77.93±1.0	79.20±2.2	79.10±1.6	78.74±0.3	79.40±1.4	79.76±2.4
GCN							
Chameleon	46.44±3.1	48.13±2.5	48.50±5.3	**51.12**±2.0	48.69±1.3	49.63±4.5	48.31±8.5
Squirrel	44.69±0.9	44.39±2.3	45.67±1.4	**46.49**±1.1	45.81±3.0	44.47±3.1	44.92±2.3
Flickr	93.16±0.8	93.49±0.8	93.33±0.3	92.83±0.6	**93.53**±0.7	93.09±0.3	93.16±0.5
Ratings	50.85±0.3	51.65±0.2	53.47±0.5	53.35±0.6	**55.41**±0.9	50.62±1.0	51.49±1.6
Arxiv-Y	52.92±0.2	52.79±0.3	50.01±0.4	53.20±0.2	**53.24**±0.1	52.69±0.7	49.55±1.0
Cora	88.28±1.2	88.83±0.7	89.01±0.7	89.01±1.6	88.28±1.4	**90.73**±1.1	88.77±0.6
PubMed	88.84±0.5	89.45±0.6	89.71±0.4	**89.85**±0.2	88.88±0.3	89.84±0.4	89.84±0.3
WikiCS	85.55±1.3	85.70±0.7	**85.78**±0.4	84.68±0.5	84.94±0.7	84.88±0.6	85.60±0.2
Citeseer	76.61±1.5	77.42±1.7	77.73±2.8	76.15±0.8	77.88±1.3	77.63±2.6	**78.08**±1.0
GAT							
Chameleon	46.63±2.5	47.94±3.2	49.44±5.4	53.18±1.3	48.88±2.0	48.69±2.3	50.56±1.7
Squirrel	46.56±1.0	45.29±2.5	45.22±3.7	46.94±1.7	46.41±1.9	45.74±0.8	44.69±1.1
Flickr	93.31±0.5	92.54±0.5	93.22±1.1	92.50±0.4	93.07±0.9	92.65±0.8	93.16±0.3
Ratings	51.94±0.7	52.07±0.5	51.02±1.6	53.56±0.3	55.72±1.7	51.30±0.9	52.31±0.7
Arxiv-Year	52.07±0.5	53.11±0.4	49.65±0.7	52.55±0.4	52.75±0.3	48.80±0.3	48.61±1.2
Cora	88.34±0.5	88.46±1.2	87.85±2.5	87.91±0.9	89.93±1.0	88.15±0.8	89.50±1.5
PubMed	88.27±1.3	90.15±0.7	89.60±0.6	89.66±0.3	89.18±0.8	89.53±0.1	89.91±0.2
WikiCS	85.78±1.0	86.12±0.5	85.25±0.6	83.92±0.6	85.36±0.4	85.79±0.9	85.46±0.8
Citeseer	76.97±1.2	76.61±2.6	77.47±1.6	76.81±2.3	77.98±1.5	76.81±2.2	76.56±1.3
ACM-GCN							
Chameleon	48.78±2.1	47.08±3.9	51.41±2.0	52.54±1.9	48.58±1.7	51.79±1.4	50.66±3.6
Squirrel	52.69±1.1	54.09±2.5	53.81±1.5	53.04±1.1	53.17±1.0	54.23±0.9	53.74±2.2
Flickr	91.82±0.9	92.39±0.8	92.30±0.6	90.98±0.7	92.19±0.3	91.99±0.7	91.62±0.9
Ratings	48.98±0.5	49.43±0.7	50.02±0.9	48.75±0.5	52.62±0.6	50.57±2.2	51.76±2.4
Arxiv-Year	44.84±1.4	47.49±1.4	44.10±1.2	45.82±1.1	47.92±1.1	44.16±0.4	43.77±0.8
Cora	87.68±0.4	86.75±1.2	90.15±0.1	89.16±0.6	88.62±1.9	88.62±0.3	89.00±0.8
PubMed	86.98±0.3	89.00±0.3	89.40±0.3	89.01±0.3	89.24±0.6	90.53±0.8	89.89±0.1
WikiCS	82.46±0.6	82.63±0.1	83.09±0.6	80.85±0.8	83.40±0.4	84.49±0.4	83.54±0.2
Citeseer	76.88±1.6	76.97±0.9	80.17±1.0	79.26±0.6	80.40±0.8	79.92±2.2	80.08±1.3

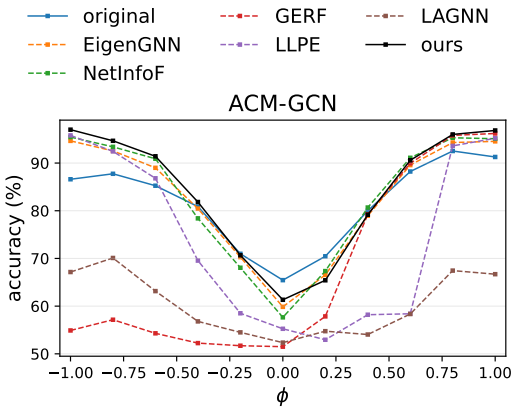


Figure 3: Node classification accuracy (%) on synthetic cSBM graph datasets for ACM-GCN, under varying homophily levels (x-axis), comparing different graph feature augmentation methods.

Table 8: Ablation of GRAPHALFA on the MLP, GAT, and ACM-GCN models. We evaluate the effect of removing individual components, reporting mean node classification accuracy (%) ± standard deviation.

Config	Chameleon	Squirrel	Ratings	Cora	Citeseer
MLP					
Original (no augmentation)	43.15±3.6	37.84±0.7	52.04±0.2	75.87±1.6	73.36±1.5
Concat $[\,\mathbf{X}_f \,\|\|\, \mathbf{X}_s\,]$ (no align.)	44.83±4.0	38.65±2.1	53.98±3.7	87.33±1.4	76.27±1.0
No struct view $\mathbf{X}_s$	44.19±1.6	39.09±2.9	52.18±1.3	85.64±0.7	77.12±1.0
Only Ego Feat., $\mathbf{X}_f = \mathbf{X}^{(0)}$	43.82±3.9	40.28±1.5	53.88±0.3	80.05±1.1	74.23±0.7
No 1-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(2)}\,]$	44.49±2.3	41.79±2.5	50.81±0.7	86.30±1.5	74.55±0.9
No 2-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(1)}\,]$	44.16±3.1	41.48±1.5	57.35±0.4	86.52±1.7	76.80±0.6
Full Framework	**48.13±1.3**	**43.57±2.3**	**57.38±0.3**	**89.20±0.9**	**78.74±0.3**
GAT					
Original (no augmentation)	41.46±4.9	36.55±1.9	48.39±0.4	85.86±0.7	73.73±2.1
Concat $[\,\mathbf{X}_f \,\|\|\, \mathbf{X}_s\,]$ (no align.)	43.48±2.8	41.57±0.7	48.06±1.0	87.77±1.4	74.70±1.4
No struct view $\mathbf{X}_s$	40.90±3.3	37.00±1.5	47.94±0.7	88.07±1.5	73.88±1.4
Only Ego Feat., $\mathbf{X}_f = \mathbf{X}^{(0)}$	42.58±6.1	38.74±2.4	48.99±0.5	87.33±0.5	77.66±1.1
No 1-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(2)}\,]$	43.26±2.7	38.70±2.2	48.52±0.6	87.11±1.1	76.41±2.3
No 2-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(1)}\,]$	43.93±4.1	40.27±1.2	50.63±1.6	88.32±1.5	75.19±1.4
No feat-label align., only $\mathcal{L}_{\text{spec}}$	44.72±3.5	42.56±3.5	50.96±1.6	87.03±0.7	76.53±1.3
No feat-struct align., only $\mathcal{L}_{\text{SC}}$	43.03±3.8	42.83±2.9	48.26±1.3	87.77±1.7	75.40±1.8
Full Framework	**48.88±2.0**	**46.41±1.9**	**55.72±1.7**	**89.93±1.0**	**77.98±1.5**
ACM-GCN					
Original (no augmentation)	42.14±1.3	46.66±5.6	51.98±2.2	86.31±2.5	78.90±0.5
Concat $[\,\mathbf{X}_f \,\|\|\, \mathbf{X}_s\,]$ (no align.)	46.52±1.2	46.12±4.8	49.85±0.8	87.19±0.9	78.39±1.0
No struct view $\mathbf{X}_s$	46.78±4.6	48.55±4.8	51.40±1.7	87.95±0.7	80.36±1.1
Only Ego Feat., $\mathbf{X}_f = \mathbf{X}^{(0)}$	48.93±4.1	49.94±2.7	46.39±2.1	87.03±1.4	77.97±1.6
No 1-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(2)}\,]$	48.59±2.7	48.68±3.4	47.54±2.4	88.05±1.2	79.81±1.7
No 2-hop, $\mathbf{X}_f = [\,\mathbf{X}^{(0)} \,\|\|\, \mathbf{X}^{(1)}\,]$	47.91±2.7	50.31±4.3	48.20±1.8	88.05±0.8	80.05±2.3
No feat-label align., only $\mathcal{L}_{\text{spec}}$	48.70±5.9	49.48±3.0	47.22±2.0	88.08±1.0	79.67±1.3
No feat-struct align., only $\mathcal{L}_{\text{SC}}$	47.34±4.4	50.31±2.8	47.79±1.8	87.68±0.4	80.30±1.4
Full Framework	**51.41±1.6**	**53.17±1.0**	**52.62±0.6**	**88.62±1.8**	**80.40±0.8**

D.3.2 COMPONENT ABLATIONS ACROSS MORE BACKBONES.

Table 8 reports ablation results for additional backbone models. Consistent with our observations in Table 2, removing any major component of GRAPHALFA leads to performance degradation across datasets.

D.3.3 STABILITY AND COST OF EMPIRICAL FREQUENCY RESPONSE ESTIMATION.

Table 9 presents the stability and overhead of estimating $\mathbf{g}_{\text{emp}}$ under different update intervals U. Across consecutive updates, GNN layers, and independent runs, the cosine similarities remain high, indicating that $\mathbf{g}_{\text{emp}}$ captures a stable propagation tendency rather than a noisy intermediate estimate. Since $\mathbf{g}_{\text{emp}}$ is computed over the full node set in a single forward pass, it does not introduce mini-batch variance. The amortized overhead is modest and decreases with larger U, showing that less frequent updates provide a favorable accuracy–cost trade-off.

Table 9: Stability and overhead of empirical frequency response estimation on Arxiv-Year with a GCN backbone. We report cosine similarity of $\mathbf{g}_{\text{emp}}$ across consecutive updates, layers, and runs, together with amortized runtime overhead. Higher similarity indicates more stable estimates.

Update interval U	Updates	Layers	Runs	Runtime (ms/epoch)	% of train runtime
$U = 5$	0.9750	0.9736	0.9916	21.24	11.87%
$U = 10$	0.9845	0.9660	0.9441	14.80	7.64%
$U = 20$	0.9673	0.9301	0.9793	4.86	3.09%

D.3.4 SENSITIVITY TO SPECTRAL SLICING HYPERPARAMETERS.

Table 10 reports sensitivity to the spectral slicing hyperparameters K, m, R, and U, varying one parameter at a time with settings $K \in \{3, 5, 7\}$, $m \in \{15, 30, 60\}$, $R \in \{2, 4, 8\}$, and $U \in \{5, 10, 20\}$. Performance is stable across configurations: larger K, m, and R can improve estimate quality but quickly saturate, while U has limited impact, consistent with the stability of $\mathbf{g}_{\mathrm{emp}}$ in Table 9. Thus, the spectral alignment module is not sensitive to precise tuning; moderate settings provide strong accuracy-cost trade-offs.

Table 10: Sensitivity analysis of spectral slicing hyperparameters on Arxiv-Year with a GCN backbone. We vary one parameter at a time and report node classification accuracy (%) as mean $\pm$ standard deviation.

Parameter	Low	Medium	High
K	51.37 ± 0.07	53.24 ± 0.13	52.78 ± 0.03
m	53.18 ± 0.36	53.24 ± 0.13	52.18 ± 0.48
R	50.65 ± 0.48	53.24 ± 0.13	52.84 ± 0.19
U	52.68 ± 0.13	53.24 ± 0.13	52.97 ± 0.02

D.3.5 LABEL-RATE SENSITIVITY.

Table 11 reports performance under reduced label rates. Although accuracy decreases as supervision becomes sparser, GRAPHALFA consistently outperforms the GCN baseline across all datasets and label rates, indicating that the alignment objectives remain useful. The observed training labels provide a subsampled estimate of the graph-level label signal; since $\mathcal{L}_{\mathrm{spec}}$ aligns normalized band-wise energy profiles via cosine similarity, it depends mainly on the relative spectral distribution of label energy rather than its absolute magnitude.

Table 11: Sensitivity to label rate with GCN. We report node classification accuracy (%) as mean $\pm$ standard deviation under different fractions of labeled training nodes.

Dataset	20% Labels		5% Labels		1% Labels	
	GCN	GRAPHALFA	GCN	GRAPHALFA	GCN	GRAPHALFA
Chamel.	39.10 ± 1.25	$\mathbf{42.60 \pm 0.71}$	37.31 ± 2.73	$\mathbf{40.33 \pm 0.62}$	32.63 ± 4.60	$\mathbf{37.30 \pm 1.00}$
Squirrel	35.48 ± 1.30	$\mathbf{39.48 \pm 2.34}$	33.93 ± 0.80	$\mathbf{36.39 \pm 0.62}$	32.90 ± 5.16	$\mathbf{35.84 \pm 1.15}$
Cora	85.13 ± 0.89	$\mathbf{85.70 \pm 0.33}$	79.25 ± 1.78	$\mathbf{81.27 \pm 1.08}$	67.07 ± 2.60	$\mathbf{69.08 \pm 1.62}$

D.3.6 ROBUSTNESS TO FEATURE AND STRUCTURAL NOISE.

Table 12 evaluates GRAPHALFA under controlled perturbations, using feature masking for feature noise and edge dropping for structural noise; we also report the average gate weight, where larger values indicate greater reliance on the attribute-derived view. GRAPHALFA consistently outperforms the base GCN under both perturbation types, and the gate adapts in the expected direction: feature masking decreases the attribute-view weight, shifting emphasis toward structure, while edge dropping increases it, reducing reliance on corrupted structural information. The results suggest that projector-gate fusion adaptively adjusts the contribution of each view according to signal reliability instead of imposing a fixed mixture.

Table 12: Robustness to feature and structural noise on Chameleon with a GCN backbone. We report node classification accuracy (%) as mean $\pm$ standard deviation. The gate weight denotes the average contribution of the attribute-derived view.

Perturbation	Rate	Base GCN	GRAPHALFA	Gate weight
None	–	42.7 ± 5.5	$\mathbf{48.7 \pm 1.3}$	0.571
Feature mask	10%	36.3 ± 4.3	$\mathbf{44.2 \pm 7.5}$	0.544
Feature mask	20%	31.1 ± 7.6	$\mathbf{42.7 \pm 1.7}$	0.534
Edge drop	5%	34.6 ± 7.8	$\mathbf{42.0 \pm 1.7}$	0.605
Edge drop	10%	32.9 ± 5.5	$\mathbf{40.6 \pm 4.9}$	0.634

D.3.7 ACCURACY ACROSS LOCAL HOMOPHILY BINS.

Table 13 reports accuracy across local homophily bins, where each node is grouped by the fraction of same-label neighbors. GRAPHALFA improves performance across most bins, with especially large gains on low- and medium-homophily nodes where message passing is more likely to mix conflicting class signals; on highly homophilic nodes, the base GCN is already near saturated, leaving little room for improvement. The augmented features also show stronger class separation, with adjusted feature homophily increasing from 0.057 to 0.290 on Arxiv-Year and from 0.010 to 0.173 on Cora, indicating that GRAPHALFA improves discriminative structure for locally challenging nodes.

Table 13: Accuracy (%) across local homophily bins with a GCN backbone. We report mean ± standard deviation.

Dataset	Local homophily	Base GCN	GRAPHALFA
Arxiv-Year	$[0, 0.25]$	39.6 ± 0.2	$\mathbf{45.4 \pm 0.5}$
	$(0.25, 0.5]$	47.1 ± 0.4	$\mathbf{56.1 \pm 0.5}$
	$(0.5, 0.75]$	49.0 ± 0.5	$\mathbf{58.5 \pm 0.3}$
	$(0.75, 1.0]$	67.0 ± 1.1	$\mathbf{77.7 \pm 1.5}$
Cora	$[0, 0.25]$	19.7 ± 6.5	$\mathbf{21.0 \pm 4.2}$
	$(0.25, 0.5]$	46.2 ± 7.4	$\mathbf{55.3 \pm 6.9}$
	$(0.5, 0.75]$	79.3 ± 4.4	$\mathbf{83.0 \pm 3.0}$
	$(0.75, 1.0]$	$\mathbf{98.5 \pm 0.7}$	98.2 ± 0.7

D.4 COMPUTATIONAL EFFICIENCY

D.4.1 PRE-PROCESSING COST OF STRUCTURE-DERIVED VIEWS.

Table 14 reports the pre-processing time required to compute different structure-derived embedding views across datasets. LINE consistently achieves the lowest or near-lowest computation time across datasets, reflecting its linear-time complexity. These results support our choice of LINE as a computationally efficient choice for instantiating the structure-derived view in our framework.

Table 14: Pre-processing runtime of structure-derived view construction: Wall-clock time (in seconds) required to compute different structure-derived embeddings.

Dataset	Node2Vec (Local)	Node2Vec (Global)	xNetMF	Role2vec	LINE	Top LPE	Top + Bottom LPE
Chameleon	3.43	3.22	0.65	6.98	0.70	1.05	0.57
Squirrel	3.99	7.18	3.31	27.03	0.81	1.50	1.40
Flickr	7.87	21.63	32.61	99.88	6.15	5.73	5.34
Ratings	19.52	47.42	9.55	249.04	5.34	26.81	15.63
Arxiv-Year	139.53	251.38	768.23	1315.86	53.50	61.76	74.64
Cora	4.84	8.63	1.06	27.56	0.16	2.62	3.73
PubMed	17.05	64.94	7.70	238.66	2.42	13.72	8.59
WikiCS	12.08	49.65	31.68	155.60	7.06	5.82	4.44
Citeseer	4.92	10.89	1.10	30.92	0.16	9.10	88.11

D.5 HYPERPARAMETER DETAILS

We consider four backbone architectures:

(1) **MLP:** We implement a two-layer MLP model with hidden dimension 256 and batch normalization. Hyperparameter search is performed over the following ranges: (i) Learning rate: [1e-4, 1e-2]; (ii) Dropout rate: [0,0.8]; (iii) Weight decay: [0, 1e-3].

(2) **GCN:** We use the implementation from (Luo et al., 2024), the hidden dimension is set to 256. Hyperparameter search is performed over the following ranges: (i) Learning rate: [5e-4, 1e-2]; (ii) Dropout rate: [0,0.8]; (iii) Weight decay: [0, 1e-3]; and (iv) Batch normalization and residual connections: {on, off}. (v) Number of layers: {2,3,4,5}.

(3) **GAT:** We use the GAT implementation from (Luo et al., 2024), with a single attention head, and the same hyperparameter search space as the GCN model.

(4) **ACM-GCN:** We use the acmgcnp variant of the ACM-GCN implementation from (Luan et al., 2022). Architectural choices follow the original implementation and are kept fixed. Hyperparameter search is performed over the following ranges: (i) Learning rate: [5e-4, 1e-2]; (ii) Dropout rate: [0,0.8]; (iii) Weight decay: [0, 1e-3].

For all models, we use the Adam optimizer. To ensure fairness, for each experimental configuration, we perform hyperparameter search via random search over 50 trials, due to the complexity and diversity of the search space. For datasets where official hyperparameters and results are provided in prior work, our reproduced baseline performance using this search procedure is comparable to reported results.

All hyperparameters associated with the proposed augmentation framework are selected via random search (jointly tuned with backbone model training end-to-end) unless otherwise specified. Specifically, they are searched over the following ranges: (i) Contrastive Loss temperature: [0.1,0.8]; (ii) λ_{SC}: [1e-2, 1]; (iii) λ_{spec}: [1e-2, 1]; (iv) λ_{reg}: [1e-4, 1]. For spectral estimation, the normalized graph Laplacian is partitioned into three fixed bands (low-, mid-, and high-frequency): [0.0,0.3), [0.3,0.7), and [0.7,2.0]. Stochastic Lanczos-based spectral slicing is performed using $R = 4$ random probes and $m = 30$ Lanczos steps.

For all structure-based node embeddings, we follow existing standard implementations and set the dimension to 256 to ensure fair comparison. For Node2Vec, we consider both local-oriented and global-oriented variants by varying the return and in-out parameters (p, q). The local variant uses walk length 25 with $p = 0.5, q = 4$, encouraging neighborhood-preserving walks, while the global variant uses walks of length 60, $p = 2$, and $q = 0.25$ to promote exploration of distant graph regions. Both variants use context size 5. For xNetMF, we set the maximum hop distance to 2, decay parameter $\alpha = 0.1$, and the number of landmarks to 10. For LINE, we use both first-order and second-order proximity with a negative sampling ratio of 5. For Laplacian Positional Encodings (LPE), we consider both the top-k variant, where the embedding only takes from the top-k eigenvectors, and the top+bottom variant, where the embedding consists of the top $k/2$ and bottom $k/2$ eigenvectors.