BEYOND HOMOPHILY: DUAL-FREQUENCY FUSION FOR MULTI-LABEL NODE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-label node classification (MLNC) predicts multiple labels for each node in graph data, where connected nodes exhibit diverse label semantics, resulting in the coexistence of homogeneous and heterogeneous neighborhood information. Considering that homogeneous information preserves semantic consistency while heterogeneous one may harm consistency, existing methods mainly suppress heterogeneous neighbors while overlooking their potential discriminability. Integrating the merits of both information empowers representations but remains two coupled issues: (1) **Feature Entanglement.** Aggregating heterogeneous neighbors mixes different label semantics, resulting in ambiguous feature distributions and obscuring decision boundaries; (2) **Optimization Inconsistency.** These entangled features induce inconsistent prediction gradients among neighbors, making it difficult to reliably assign the contribution of neighborhood information. To mitigate these issues, we propose a Dual-Frequency Fusion Network (DF^{2}Net), which realizes effective fusion of homogeneous and heterogeneous neighborhood information from a frequency perspective. Specifically, DF^{2}Net models homogeneous and heterogeneous information as low- and high-frequency signals, respectively, and develops a dual-frequency fusion strategy to exploit their complementarity. Moreover, DF^{2}Net introduces a Joint Representation-Optimization Consistency (JROC) module containing Label-guided Feature Refinement (LFR) and Gradient Consistency Calibration (GCC) to align representation and optimization spaces in a mutually reinforcing manner. LFR aligns node representations with relevant labels to disentangle ambiguous features and clear decision boundaries, yielding reliable prediction gradients. Based on the similarities of these gradients, GCC realizes to quantify optimization consistency and calibrate the fusion process. Through this synergy, DF^{2}Net successfully exploits complementary homogeneous and heterogeneous information. Extensive experiments show that DF^{2}Net achieves superior performance over state-of-the-art MLNC methods.

1 INTRODUCTION

Single-label node classification (SLNC) aims to assign a specific label to each node in graph data (Zhong et al., 2025). However, many real-world applications contain nodes associated with multiple labels due to their diverse properties (Zhou et al., 2021). For example, proteins in protein-protein interaction networks usually correspond to multiple gene functions (Gao et al., 2023), which naturally leads to the more challenging multi-label node classification (MLNC) problem. A fundamental characteristic of MLNC is the inherent heterogeneity of multi-label graphs, where nodes with different label sets are frequently connected, resulting in complex neighborhood interactions (Bei et al., 2025). Recent MLNC methods mainly focus on mitigating the adverse effects of heterogeneous interactions. LIP (Sun et al., 2025) introduces a global loss weighting mechanism to alleviate optimization uncertainty induced by heterogeneous label semantics. CorGCN (Bei et al., 2025) constructs label-specific correlation graphs through graph decomposition to avoid direct aggregation among heterogeneous nodes. Despite their effectiveness, these methods generally regard heterogeneous neighbors as undesirable interference and overlook their potential discriminability.

To further investigate how neighborhood information influences model performance, we conduct a comparative analysis on the Ogbn-Proteins dataset (Hu et al., 2020). Specifically, we train a graph convolutional network (GCN) (Kipf & Welling, 2017) with the topology and compare it with a

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.

topology-free baseline by setting the adjacency matrix to the identity matrix. To characterize the effect of topology information during the optimization process, we extract the prediction gradients of both models *w.r.t.* logits and calculate the similarity between them. A higher similarity indicates that neighborhood aggregation preserves the optimization direction, whereas a lower similarity reflects the deviations induced by heterogeneous neighbors. Meanwhile, we quantify the gains of topology information using ΔAP, defined as the Average Precision Difference between the GCN and the topology-free baseline, and the correlation between gradient similarity and ΔAP is presented in Figure 1. The results reveal two observations: (1) *Neighborhood topology provides complementary information, leading to performance gains.* (2) *The benefit of topology information varies with the prediction gradient variation, where moderate deviations yield larger improvements than excessive or negligible changes.* These observations indicate that the complementary topological information manifests through two distinct structural contexts: (1) Negligible gradient changes correspond to homogeneous contexts that preserve semantic consistency, while (2) gradient deviations reflect heterogeneous contexts introducing potential discriminability. Considering that both consistent semantics and moderate structural deviations contribute to the overall performance gains, an ideal model is expected to integrate the merits of homogeneous and heterogeneous neighborhood information.

In this paper, we characterize neighborhood information from a frequency perspective to mine the merits, where homogeneous neighbors are interpreted as low-frequency signals that preserve smooth semantics, while heterogeneous neighbors are regarded as high-frequency signals capturing discriminability. However, directly integrating merits of these complementary signals remains challenging due to two ingrained and coupled issues: (1) **Feature Entanglement.** Aggregating heterogeneous neighbors mixes representations associated with different labels, resulting in ambiguous feature distributions and obscuring decision boundaries; (2) **Optimization Inconsistency.** These entangled representations induce inconsistent prediction gradients among connected nodes, making it difficult to reliably optimize the contribution of neighborhood information.

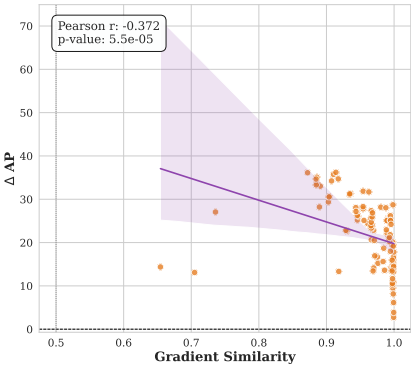


Figure 1: Correlation analysis between gradient similarity and performance. The scatter points represent individual labels, the purple line denotes the linear regression trend, and the shaded area indicates the confidence interval. The Pearson coefficient r alongside a small p-value demonstrates a statistically significant negative correlation.

To alleviate these issues, we propose a novel MLNC framework, named Dual-Frequency Fusion Network (DF^{2}Net). DF^{2}Net first introduces a *Dual-Frequency Representation Extraction* strategy to transform neighborhood information into low- and high-frequency contexts, capturing smooth semantics and discriminative differences simultaneously. The predictions from the two frequency channels are then adaptively fused to exploit their complementary information. Furthermore, to establish a consistent learning process across representation and optimization spaces, we introduce a *Joint Representation-Optimization Consistency (JROC)* module, which progressively improves representation quality and regulates the contribution of neighborhood signals through two interconnected strategies. First, the *Label-guided Feature Refinement (LFR)* strategy utilizes supervision to refine node representations. By pulling nodes toward relevant labels and pushing them away from irrelevant ones, LFR alleviates feature entanglement and improves the discriminability of representations. Based on these refined representations, the *Gradient Consistency Calibration (GCC)* strategy measures the similarity between prediction gradients of connected nodes and leverages this metric to further calibrate the fusion process of the frequency signals. In this way, GCC reduces the optimization inconsistency induced by heterogeneous neighborhoods. Through this representation-to-optimization learning process, LFR provides more reliable representations and prediction gradients for GCC, while GCC further regulates the assignments of neighborhood information according to optimization consistency. Together, these strategies enable DF^{2}Net to exploit complementary homogeneous and heterogeneous information while mitigating feature entanglement and optimization inconsistency. The main contributions are summarized as follows:

(1) We revisit neighborhood information in multi-label graphs and reveal that heterogeneous interactions also provide beneficial signals rather than mere negative effects.

(2) We propose DF^{2}Net, a dual-frequency graph learning framework that adaptively integrates low- and high-frequency neighborhood information for MLNC.

(3) We develop the JROC module to jointly improve representation discriminability and optimization consistency through LFR and GCC strategies. Extensive experiments on multiple datasets demonstrate the superiority of DF^{2}Net over existing state-of-the-art methods.

2 Related Work

Single-Label Node Classification. Graph neural networks (GNNs) have achieved remarkable success in the SLNC task by aggregating neighborhood information. To address the challenges of structural noise and data redundancy in graphs, considerable methods have been proposed. For instance, methods optimize graph structures dynamically through information bottlenecks (Zhong et al., 2025), leverage structural entropy (Duan et al., 2024), or utilize contrastive learning to jointly mitigate graph and label noise (Zhu et al., 2024). To further enhance model robustness, researchers attempt to align feature and semantic spaces via dual information bottlenecks (Zhong et al., 2024), or introduce periodic projections to alleviate over-smoothing in deep neural network architectures (Chen et al., 2024b). Additionally, advanced graph augmentation techniques, such as adversarial edge dropping (Chen et al., 2025) and graph mixup (Lu et al., 2025), have been developed to improve generalization. Despite their effectiveness in SLNC, these methods generally overlook the diverse label semantics present in real-world data. Consequently, they are not suitable for more complex scenarios where nodes exhibit multiple labels, highlighting the necessity of exploring MLNC.

Multi-Label Learning. Multi-label learning aims to learn a predictive model from data where each instance is associated with multiple labels (Wei et al., 2025), capturing both the instance-label mappings and their intricate correlations (Wu et al., 2021). While early approaches like binary relevance (Tsoumakas & Katakis, 2007) often yield suboptimal performance due to the ignorance of strong dependencies (Bei et al., 2025), recent deep learning models have successfully shifted towards mining complex nonlinear label relationships (Lyu et al., 2022; 2025). Beyond modeling complex label correlations, recent research has also expanded to handle increasingly complex input feature spaces. Specifically, when instances are described by multiple view data, multi-view multi-label classification has been proposed to integrate heterogeneous representations through sophisticated fusion strategies. Recent advancements include fusion frameworks that integrate complementary feature and decision strengths (Lin et al., 2026), and hypergraph neural networks designed to adaptively capture group-level semantic similarities (Shan et al., 2026). Furthermore, to handle missing views and unannotated labels, researchers have introduced robust mechanisms such as uncertainty-driven dynamic fusion (Wen et al., 2026) and label-driven contrastive learning (Liu et al., 2025). Despite the significant progress in general multi-label learning, its application tailored for graph-structured data remains relatively underexplored, which is the focus of this work.

Multi-Label Node Classification. Multi-label node classification (MLNC) focuses on assigning multiple relevant labels to individual nodes, where the primary challenge lies in modeling complex node-label correlations. To address this, pioneering methods like ML-GCN (Gao et al., 2019) and MuLGCN (Shi et al., 2022) construct node-label and label-label relationships to capture complementary semantics, while LANC (Zhou et al., 2021) and SMLG (Song et al., 2021) further optimize representations by exploiting topological structures. Recent progress extends MLNC to more scenarios, including few-shot learning (Xiao et al., 2022), label co-occurrence interference (Sun et al., 2025), and large-scale graph scalability (Chen et al., 2024a; Yu et al., 2024). Despite showing advancements, most approaches employ uniform message-passing mechanisms that indiscriminately mix homogeneous and heterogeneous neighborhood information, resulting in semantic entanglement and ambiguous representations. Although CorGCN (Bei et al., 2025) attempts to mitigate this issue via label-specific graph decomposition, it treats different label semantics separately and lacks an effective mechanism to exploit their complementary information. However, heterogeneous interactions cannot be regarded as purely harmful, as they may contain discriminative information beneficial for prediction. This motivates the development of a unified framework that jointly mines complementary homogeneous and heterogeneous information.

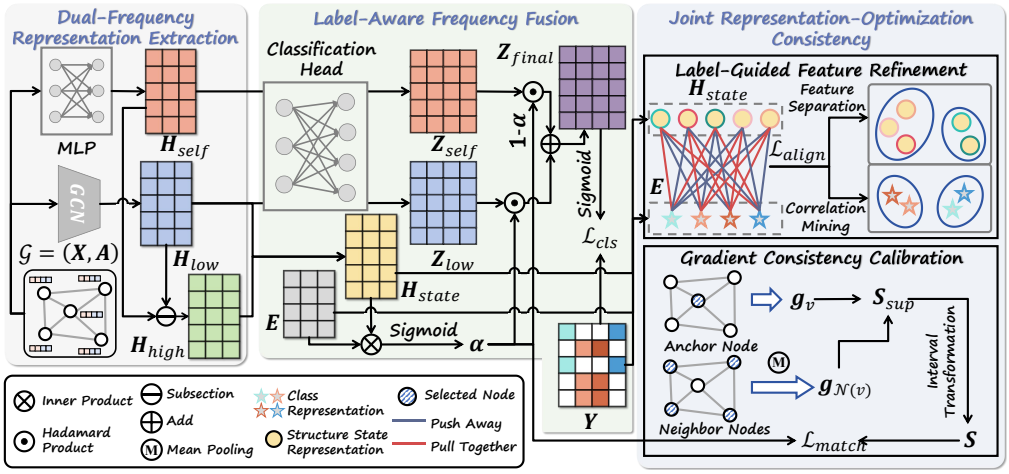


Figure 2: The architecture of the proposed framework. The Dual-Frequency Representation Extraction module extracts intrinsic representation $\boldsymbol{H}_{self}$, low-frequency structure context $\boldsymbol{H}_{low}$, and high-frequency structure context $\boldsymbol{H}_{high}$. The Label-Aware Frequency Fusion module combines the structure state representation $\boldsymbol{H}_{state}$ and label representation $\boldsymbol{E}$ to generate the fusion weight $\boldsymbol{\alpha}$, which integrates the predictions $\boldsymbol{Z}_{self}$ and $\boldsymbol{Z}_{low}$ into the final output $\boldsymbol{Z}_{final}$. To ensure optimization consistency, the Gradient Consistency Calibration module computes a similarity score $\boldsymbol{S}$ based on the prediction gradients of the anchor node $\boldsymbol{g}_v$ and its aggregated neighbor gradient $\boldsymbol{g}_{\mathcal{N}(v)}$, establishing a matching loss $\mathcal{L}_{match}$ to adjust $\boldsymbol{\alpha}$. Concurrently, the Label-Guided Feature Refinement module conducts correlation mining between $\boldsymbol{H}_{state}$ and $\boldsymbol{E}$ with an alignment loss $\mathcal{L}_{align}$.

3 Methodology

In this section, we present the details of DF^{2}Net. As illustrated in Figure 2, the architecture decouples representation learning into two complementary streams to capture homogeneous and heterogeneous contexts, which are then dynamically integrated through an adaptive fusion mechanism. To mitigate the entanglement of the feature distributions, a Label-guided Feature Refinement strategy is adopted. Additionally, the fusion process is further regularized by a Gradient Consistency Calibration strategy to balance the homogeneous and heterogeneous contexts appropriately.

3.1 Preliminary

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a multi-label graph, consisting of a node set $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ with N nodes and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ capturing their pairwise topological connections. We define the adjacency matrix as $\boldsymbol{A} \in \{0,1\}^{N \times N}$, where $\boldsymbol{A}_{i,j} = 1$ implies a link between v_i and v_j, and 0 otherwise. The corresponding node attributes are parameterized by $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, with $\boldsymbol{x}_i$ signifying the feature vector of v_i. Each node $v_i \in \mathcal{V}$ maps to a specific subset of labels $\boldsymbol{y}_i \subseteq \mathcal{Y}$, with $\mathcal{Y}$ encompassing all available categories and $L = |\mathcal{Y}|$ acting as the total label count. Formally, $\boldsymbol{y}_i$ is encoded into a multi-hot vector $\boldsymbol{y}_i = [\boldsymbol{y}_{i,1}, \boldsymbol{y}_{i,2}, \ldots, \boldsymbol{y}_{i,L}]$, such that $\boldsymbol{y}_{i,j} = 1$ if the j-th label is assigned to v_i, and $\boldsymbol{y}_{i,j} = 0$ if it is absent. Ultimately, provided with the structural information $\boldsymbol{A}$, the feature matrix $\boldsymbol{X}$, and a subset of labeled nodes, our objective is to learn a predictive function $f : \mathcal{V} \times \mathcal{E} \to \{0,1\}^{N \times L}$ to infer the correct label assignments for unlabeled nodes.

3.2 Dual-Frequency Representation Extraction

In general, homogeneous contexts reflect consensus in line with the anchor node, corresponding to low-frequency signals, while the discrepancy incurred by heterogeneous contexts acts as high-frequency signals. We introduce the following theorems for guiding DF^{2}Net to extract both homogeneous and heterogeneous contexts:

Theorem 3.1 (Low-Frequency Property of GCN) *The neighborhood aggregation in GCN acts as a low-pass filter, which inherently preserves low-frequency signals.*

Theorem 3.2 (High-Frequency Property of Discrepancy) *The discrepancy between an anchor node and its neighborhoods acts as a high-pass filter and captures high-frequency signals.*

The proofs of the two theorems are provided in **Appendix** A.

(1) **Low-Frequency Homogeneous Stream.** We employ a GCN to extract homogeneous contexts since this operation inherently acts as a low-pass filter, smoothing the node features over the local topology. The extraction process of the low-frequency representation $\boldsymbol{H}_{low}$ is defined as:

$$\boldsymbol{H}_{low} = \text{GCN}(\boldsymbol{X}, \boldsymbol{A}), \tag{1}$$

where $\boldsymbol{X}$ is the input feature matrix and $\boldsymbol{A}$ is the adjacency matrix.

(2) **High-Frequency Heterogeneous Stream.** The heterogeneous high-frequency information reflects the discrepancy between the anchor node and its neighbors. To capture this, we implement a Multi-Layer Perceptron (MLP) on the raw attributes to produce an intrinsic representation $\boldsymbol{H}_{self}$ with the same dimension as $\boldsymbol{H}_{low}$. Since $\boldsymbol{H}_{low}$ captures the low-frequency neighbor information, the difference between $\boldsymbol{H}_{self}$ and $\boldsymbol{H}_{low}$ reflects the discrepancy between the intrinsic self information and the neighbor information. From a frequency perspective, this discrepancy corresponds to the high-frequency signal, and it is defined as:

$$\boldsymbol{H}_{self} = \text{MLP}(\boldsymbol{X}), \;\; \boldsymbol{H}_{high} = \boldsymbol{H}_{self} - \boldsymbol{H}_{low}. \tag{2}$$

Discrepancy representation $\boldsymbol{H}_{high}$ encapsulates the high-frequency signals where the local neighbors show discrepancy against the anchor node, supplementing the representation discriminability.

3.3 Label-Aware Frequency Fusion

Considering that nodes exhibit individual local structure properties, their demands for low- and high-frequency signals vary significantly. To achieve complementary predictions between these signals for each node, we introduce an adaptive fusion mechanism that captures the correlation between structure contexts and specific labels to dynamically adjust the fusion process.

The fusion process is based on the structure context demands for nodes, which encourages nodes to appropriately balance the neighbor signals and the nodes themselves. To explore the structure context demands, we concatenate the low-frequency representation $\boldsymbol{H}_{low}$ and the high-frequency representation $\boldsymbol{H}_{high}$, processing them through an MLP to generate a structure state representation $\boldsymbol{H}_{state}$ that assimilates both local homophily and heterogeneity as

$$\boldsymbol{H}_{state} = g(\boldsymbol{H}_{low} \parallel \boldsymbol{H}_{high}), \tag{3}$$

where $g(\cdot)$ denotes the MLP, and $\parallel$ represents the concatenation operator. We then compute the inner product between the structural state $\boldsymbol{H}_{state}$ and a learnable label representation matrix $\boldsymbol{E} \in \mathbb{R}^{L \times d}$ to evaluate the correlation between them as

$$\boldsymbol{\alpha} = \sigma(\boldsymbol{H}_{state}\boldsymbol{E}^{\top}), \tag{4}$$

where $\sigma(\cdot)$ denotes the Sigmoid function, and $\boldsymbol{\alpha}$ represents the adaptive fusion weight.

The final prediction is an integration of the low- and high-frequency components. Rather than directly utilizing the high-frequency representation $\boldsymbol{H}_{high}$, we alternatively employ the intrinsic representation $\boldsymbol{H}_{self}$ with the low-frequency representation $\boldsymbol{H}_{low}$. Since both $\boldsymbol{H}_{self}$ and $\boldsymbol{H}_{low}$ are the elements used to calculate $\boldsymbol{H}_{high}$, the direct fusion between $\boldsymbol{H}_{high}$ and $\boldsymbol{H}_{low}$ requires repetitive memory for $\boldsymbol{H}_{self}$ and $\boldsymbol{H}_{low}$, which introduces unnecessary computational overhead. In view of this, we directly apply a shared classification head to $\boldsymbol{H}_{low}$ and $\boldsymbol{H}_{self}$ with the parameter $\boldsymbol{W}$, which yields the corresponding predictions as $\boldsymbol{Z}_{low} = \boldsymbol{H}_{low}\boldsymbol{W}$ and $\boldsymbol{Z}_{self} = \boldsymbol{H}_{self}\boldsymbol{W}$, respectively. We then implement the fusion weight $\boldsymbol{\alpha}$ to balance these classification results:

$$\boldsymbol{Z}_{final} = \boldsymbol{\alpha} \odot \boldsymbol{Z}_{low} + (\boldsymbol{1} - \boldsymbol{\alpha}) \odot \boldsymbol{Z}_{self}, \tag{5}$$

where $\odot$ denotes the Hadamard product. This strategy avoids the redundant computation caused by $\boldsymbol{H}_{high}$, and achieves the same effect as directly weighing the low- and high-frequency components according to the following equation:

$$\boldsymbol{Z}_{high} = \boldsymbol{H}_{high}\boldsymbol{W} = (\boldsymbol{H}_{self} - \boldsymbol{H}_{low})\boldsymbol{W} = \boldsymbol{Z}_{self} - \boldsymbol{Z}_{low}. \tag{6}$$

Substituting Eq. (6) into Eq. (5) yields the equivalent transformation:

$$\boldsymbol{Z}_{final} = \boldsymbol{Z}_{low} + (1 - \boldsymbol{\alpha}) \odot \boldsymbol{Z}_{high}. \tag{7}$$

This equation demonstrates that dynamically weighing $\boldsymbol{Z}_{low}$ and $\boldsymbol{Z}_{self}$ also realizes a trade-off between low- and high-frequency components. Finally, we employ the Binary Cross-Entropy (BCE) loss (Tu et al., 2026) to optimize the fusion process with the ground-truth of training samples $\tilde{\boldsymbol{Y}}$:

$$\mathcal{L}_{freq} = \text{BCE}(\sigma(\boldsymbol{Z}_{final}), \tilde{\boldsymbol{Y}}). \tag{8}$$

3.4 Joint Representation-Optimization Consistency

So far, we have been able to utilize the complementary low- and high-frequency signals in the multi-label graph, yet two issues caused by the heterogeneity, namely **Feature Entanglement** and **Optimization Inconsistency**, still harm the model performance. To mitigate the harmfulness, we introduce the *Label-Guided Feature Refinement* and *Gradient Consistency Calibration* strategies.

3.4.1 Label-guided Feature Refinement

The loss $\mathcal{L}_{freq}$ optimizes prediction probabilities for labels independently, yet lacks a driving force to push irrelevant classes away and mitigate feature entanglement. To this end, we promote the alignment between the node and the label representations with an alignment loss:

$$\mathcal{L}_{align} = -\frac{1}{|\mathcal{V}_{train}|} \sum_{v \in \mathcal{V}_{train}} \frac{1}{|\mathcal{Y}_v|} \sum_{i \in \mathcal{Y}_v} \log \frac{\exp(\boldsymbol{h}_{state}^{(v)} \cdot \boldsymbol{e}_i / \tau)}{\sum_{j \in \mathcal{Y}} \exp(\boldsymbol{h}_{state}^{(v)} \cdot \boldsymbol{e}_j / \tau)}, \tag{9}$$

where τ is a temperature coefficient, and $\boldsymbol{h}_{state}^{(v)}$ is the state representation of node v. $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$ are the i-th and j-th rows of the label representation $\boldsymbol{E}$, corresponding to the representations of the i-th and j-th classes. $\mathcal{V}_{train}$ is the training set, $\mathcal{Y}$ denotes the label set, and $\mathcal{Y}_v$ is the label set of node v.

Unlike the loss $\mathcal{L}_{freq}$ treating each label independently, this alignment loss calculates the similarities between the anchor node and all labels simultaneously, providing two synergistic merits: (1) **Feature Separation.** By forcing node representations to align with their labels while pushing them away from irrelevant ones, this mechanism utilizes the supervision to separate the entangled features and mitigates the entanglement caused by heterogeneous neighbors. (2) **Label Correlation Mining.** Through closing the node and all its relevant labels, the inherent correlations and dependencies among different labels, such as co-occurrence, are naturally mined. Based on these two merits, the model generates more expressive representations for more accurate predictions, which leads to high-quality predictive gradients for realizing Gradient Consistency Calibration.

3.4.2 Gradient Consistency Calibration

The loss $\mathcal{L}_{freq}$ neglects the impact of neighbor heterogeneity. When aggregating information from heterogeneous neighbors, the optimization gradient directions between the neighbors and the anchor node often show inconsistency. To detect such inconsistency, we employ an auxiliary classifier to generate predictive gradients, revealing whether the conflicting optimization directions exist in neighbors. We introduce the predictive gradient similarity as supervision to guide the adaptation of the fusion weight $\boldsymbol{\alpha}$, where this supervision calculation is based on three components:

(1) **Prediction Gradient.** We introduce an auxiliary classifier to compute the predictive error $\boldsymbol{g}_v = P_{aux}(v) - \boldsymbol{Y}_v$ for each node v independently, where $P_{aux}(v)$ is the prediction of an auxiliary classifier for node v, $\boldsymbol{Y}_v$ is the label of node v, and $\boldsymbol{g}_v$ serves as the predictive gradient.

Theorem 3.3 *Let $\mathcal{L}_{aux}$ be the classification loss for training the auxiliary classifier, then the prediction error $\boldsymbol{g}_v = P_{aux}(v) - \boldsymbol{Y}_v$ is equivalent to the gradient of $\mathcal{L}_{aux}$ w.r.t. the logits.*

The proof is provided in **Appendix** A.

(2) **Aggregated Neighbor Gradient.** To estimate the joint optimization direction of the local neighbors, we calculate the aggregated gradients of neighbors as

$$\boldsymbol{g}_{\mathcal{N}(v)} = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} \boldsymbol{g}_u, \tag{10}$$

where $\mathcal{N}(v)$ denotes the neighbor set of the node v, $\boldsymbol{g}_u$ is the gradient of node u, and $\boldsymbol{g}_{\mathcal{N}(v)}$ is the aggregated neighbor gradient.

(3) **Calculation of Supervision.** For an anchor node v, we quantify the optimization consistency between the anchor node and its neighborhoods. Specifically, we compute the similarity between the anchor node's prediction gradient $\boldsymbol{g}_v$ and its aggregated neighbor gradient $\boldsymbol{g}_{\mathcal{N}(v)}$:

$$\boldsymbol{S}_{sup}(v) = \frac{\boldsymbol{g}_v \odot \boldsymbol{g}_{\mathcal{N}(v)}}{|\boldsymbol{g}_v||\boldsymbol{g}_{\mathcal{N}(v)}| + \epsilon}, \tag{11}$$

where $|\cdot|$ represents the element-wise absolute value, and ϵ is a small positive value. A prediction gradient indicates the optimization direction to minimize the prediction error. Thus, $\boldsymbol{S}_{sup}(v)$ can quantify the optimization consistency between an anchor node v and its neighbors for all the labels.

To guide the determination of the label-aware fusion weight $\boldsymbol{\alpha}$, we transform $\boldsymbol{S}_{sup}(v)$ into the interval $[0, 1]$, formulating the final similarity score vector as:

$$\boldsymbol{S}(v) = \frac{\boldsymbol{S}_{sup}(v) + \mathbf{1}}{2}. \tag{12}$$

Here, $\boldsymbol{S}(v)$ acts as a measurement for the optimization consistency. An element value approaching 1 in $\boldsymbol{S}(v)$ signifies a consistent optimization direction, indicating that the local neighborhood consensus is reliable and suggesting an enhanced confidence in the low-frequency information. Conversely, a value approaching 0 exposes optimization inconsistency, suggesting to suppress the propagation of heterogeneous neighbor information for the specific label.

Based on this, $\boldsymbol{S}(v)$ naturally functions as a supplementary supervision signal to calibrate the fusion weight. In view of this, we construct a matching loss function as:

$$\mathcal{L}_{match} = \frac{1}{|\mathcal{V}_{train}|} \sum_{v \in \mathcal{V}_{train}} \|\boldsymbol{\alpha}_v - \boldsymbol{S}(v)\|_2^2, \tag{13}$$

where $\boldsymbol{\alpha}_v$ is the fusion weight selected from $\boldsymbol{\alpha}$ for node v. With this constraint, the model adaptively adjusts the fusion weight $\boldsymbol{\alpha}_v$, thereby facilitating a more reliable fusion process.

3.5 Overall Loss Function

We formulate the overall loss function by integrating the aforementioned components, namely the classification losses $\mathcal{L}_{freq}$ and $\mathcal{L}_{aux}$, the alignment loss $\mathcal{L}_{align}$, and the matching loss $\mathcal{L}_{match}$. The total loss is defined as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{align} + \lambda_2 \mathcal{L}_{match}, \tag{14}$$

where λ_1 and λ_2 are penalty coefficients, and $\mathcal{L}_{cls} = \mathcal{L}_{freq} + \mathcal{L}_{aux}$. By optimizing this loss, the DF^{2}Net refines representation discriminability and optimization consistency in a mutually reinforcing manner, thereby effectively exploiting both homogeneous and heterogeneous contexts.

4 Experiments

In this section, we evaluate our proposed DF^{2}Net on five public multi-label graph datasets, including four biological graph datasets (*i.e.*, **Humloc**, **PCG**, **PPI**, and **Ogbn-Proteins**), and a Wikipedia network (*i.e.*, **Delve**). In our experiments, we compare our method with six representative MLNC methods, including **MLGW** (Akujuobi et al., 2019), **ML-GCN** (Gao et al., 2019), **LANC** (Zhou et al., 2021), **SMLG** (Song et al., 2021), **LIP** (Sun et al., 2025), and **CorGCN** (Bei et al., 2025). For evaluating the model performance between our proposed model and comparative models, we adopt seven metrics, namely Ranking Loss, Hamming Loss, Macro-AUC, Micro-AUC, Macro-AP, Micro-AP, and label ranking AP. All experiments are implemented in PyTorch and conducted on a server with NVIDIA GeForce 3090.

4.1 Multi-Label Node Classification

In this experiment, we evaluate the performance of DF^{2}Net against six baselines. To ensure stable results, we run all experiments five times with different seeds. We report the means and standard deviations of the metrics in Table 1. According to the results, we have the following observations:

Table 1: Comparisons with multi-label node classification results. The best results are highlighted in **red**, and the second-best results are highlighted with underline.

Datasets	Metrics	MLGW	ML-GCN	LANC	SMLG	LIP	CorGCN	**Ours**
Humloc	Ranking ↓	13.99±0.62	13.28±0.70	14.66±0.36	17.82±0.43	**12.31±0.14**	12.57±0.31	12.54±0.19
	Hamming ↓	8.15±0.04	7.46±0.22	7.82±0.16	8.55±0.26	12.04±0.66	7.37±0.17	**7.29±0.07**
	Ma-AUC ↑	71.40±2.74	72.41±2.47	68.73±1.76	72.64±1.65	73.24±1.98	**77.31±1.58**	74.24±0.72
	Mi-AUC ↑	86.29±0.58	87.63±0.52	86.06±0.47	79.06±0.38	85.49±0.45	88.57±0.37	**88.76±0.10**
	Ma-AP ↑	20.94±0.81	23.78±1.17	20.94±1.69	21.17±0.61	27.47±2.44	26.91±1.67	**28.39±0.77**
	Mi-AP ↑	40.63±1.56	47.76±1.59	42.10±2.61	35.53±0.18	40.45±1.17	48.97±1.30	**54.69±0.55**
	LRAP ↑	61.64±1.51	64.91±1.14	61.02±1.45	54.27±0.38	64.21±0.79	65.40±0.83	**68.46±0.45**
PCG	Ranking ↓	29.36±0.51	27.11±0.44	28.26±0.51	28.15±0.56	29.22±0.45	25.97±0.57	**25.37±0.30**
	Hamming ↓	13.09±0.14	12.56±0.31	12.59±0.18	16.28±0.16	20.96±1.36	**12.41±0.23**	12.68±0.05
	Ma-AUC ↑	53.45±1.45	61.48±0.84	58.75±1.05	63.02±0.97	63.34±1.10	64.86±1.05	**65.29±0.24**
	Mi-AUC ↑	69.16±0.43	72.03±0.40	71.31±0.56	71.08±0.62	70.12±0.42	74.16±0.61	**74.32±0.19**
	Ma-AP ↑	15.68±0.71	20.95±0.91	19.49±0.85	22.73±0.75	23.17±0.86	24.64±0.90	**25.31±0.34**
	Mi-AP ↑	24.06±0.43	29.33±1.21	27.54±0.98	24.77±0.85	26.89±0.37	31.91±1.07	**34.62±0.08**
	LRAP ↑	44.52±0.25	48.21±0.57	46.62±1.27	47.58±0.86	45.06±1.03	49.04±0.83	**51.38±0.38**
PPI	Ranking ↓	24.38±0.20	20.05±0.62	18.44±0.14	18.38±0.30	23.25±0.36	16.17±0.22	**14.11±0.23**
	Hamming ↓	24.38±0.20	23.83±0.14	21.85±0.28	31.06±0.32	30.66±0.01	20.79±0.29	**19.33±0.37**
	Ma-AUC ↑	61.11±0.45	70.26±1.12	74.24±0.42	74.04±0.41	77.18±0.11	77.54±0.42	**80.73±0.36**
	Mi-AUC ↑	73.91±0.22	78.37±0.72	81.15±0.26	80.46±0.26	77.33±0.30	83.35±0.29	**85.49±0.24**
	Ma-AP ↑	40.81±0.43	50.69±1.54	57.19±0.15	55.71±0.70	59.70±0.08	61.33±0.67	**65.55±0.63**
	Mi-AP ↑	59.05±0.27	64.98±1.32	69.59±0.27	63.11±0.62	62.04±0.36	72.35±0.58	**75.72±0.50**
	LRAP ↑	66.14±0.32	66.68±1.01	68.87±0.35	63.68±0.61	62.55±0.29	71.40±0.34	**76.62±0.46**
Ogbn-Proteins	Ranking ↓	25.28±0.03	15.09±0.19	25.12±0.12	15.03±0.18	27.77±0.30	10.61±0.42	**7.40±0.15**
	Hamming ↓	17.71±0.05	12.40±0.71	17.58±0.09	11.37 ± 0.04	33.58±0.28	10.63±0.09	**8.71±0.04**
	Ma-AUC ↑	50.03±0.11	64.91±0.52	57.65±0.92	68.20±0.01	74.33±0.34	79.71±0.86	**87.93±0.12**
	Mi-AUC ↑	69.43±0.04	72.11±0.33	70.95±0.22	53.98±0.12	73.92±0.29	82.93±0.70	**89.71±0.11**
	Ma-AP ↑	19.14±0.02	17.45±0.30	23.31±0.38	11.54±0.09	39.41±0.73	32.73±1.65	**51.42±0.43**
	Mi-AP ↑	41.14±0.20	29.17±0.33	42.69±0.02	24.08±0.04	40.54±0.63	42.86±1.43	**60.35±0.36**
	LRAP ↑	52.88±0.15	72.44±0.01	53.16±0.06	71.95±0.30	41.67±0.24	76.69±0.42	**81.76±0.20**
Delve	Ranking ↓	2.06±0.11	3.33±0.01	2.76±0.03	5.07±0.02	3.17±0.01	2.41±0.02	**1.80±0.07**
	Hamming ↓	2.73±0.23	2.99±0.01	3.62±0.07	7.50±0.01	7.39±0.02	2.48±0.01	**1.77±0.01**
	Ma-AUC ↑	95.25±0.34	95.71±0.04	93.89±0.31	95.26±0.09	96.70±0.02	96.84±0.10	**97.41±0.04**
	Mi-AUC ↑	97.45±0.33	97.29±0.02	96.69±0.14	95.11±0.04	97.37±0.04	98.05±0.02	**98.53±0.06**
	Ma-AP ↑	70.02±0.63	65.20±0.07	63.53±0.34	65.67±0.31	70.10±0.86	72.80±0.26	**77.12±0.68**
	Mi-AP ↑	88.66±0.76	81.78±0.09	82.25±0.28	68.97±0.25	80.19±0.02	86.12±0.03	**90.65±0.11**
	LRAP ↑	91.28±0.38	86.12±0.05	87.23±0.13	76.47±0.27	86.53±0.03	89.20±0.01	**93.10±0.07**

Table 2: Friedman statistics τ_F on all metrics with corresponding p-value.

Metrics	Ranking Loss	Hamming Loss	Macro-AUC	Micro-AUC	Macro-AP	Micro-AP	LRAP
τ_F	15.8571	23.1429	27.4286	21.7714	24.4071	23.3143	20.8286
p-value	0.0145	0.0008	0.0001	0.0013	0.0004	0.0007	0.0020

(1) DF^{2}Net consistently exhibits superior performance compared with all baseline models across nearly all metrics and datasets. Notably, on the PPI, Ogbn-Proteins, and Delve datasets, DF^{2}Net achieves the best results across all evaluation metrics and shows substantial improvements.

(2) Methods that neglect to address heterogeneous information during neighborhood aggregation, such as MLGW, LANC, and SMLG, easily mix ambiguous semantic signals and lead to feature entanglement. When compared with them, DF^{2}Net and CorGCN show significantly better performance since they consider the necessity of tackling heterogeneous and homogeneous information.

(3) In comparison with recent advanced models like LIP and CorGCN, which attempt to address heterogeneity, DF^{2}Net decouples neighborhood signals from a frequency perspective and employs an adaptive fusion strategy. This mechanism realizes an effective fusion process, ensuring that the model exploits the discriminative high-frequency information and the consistent low-frequency information in a mutually reinforcing manner.

In addition, we conduct a Friedman test (Demšar, 2006) and a Bonferroni-Dunn test (Demšar, 2006) to reveal the statistical significance among the comparative algorithms, which are shown in Table 2 and Figure 3, respectively. The Friedman test reveals that the null hypothesis of indistinguishable performance among all comparative algorithms is rejected at a 0.05 significance level, and the Bonferroni-Dunn test illustrates that DF^{2}Net has the best rank across all metrics, establishing a significant statistical advantage against the baselines.

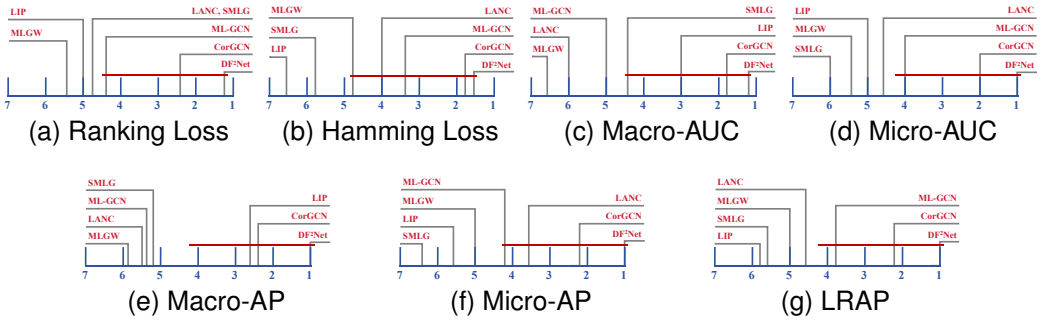


Figure 3: Bonferroni-Dunn test between DF^{2}Net and other comparative algorithms (CD = 3.27 at 0.05 significance level).

Table 3: Ablation study between DF^{2}Net and three variants.

HumLoc $\mathcal{L}_{align}$	HumLoc $\mathcal{L}_{match}$	HL ↓	RL ↓	Ma-AP ↑	Mi-AP ↑	Ma-AUC ↑	Mi-AUC ↑	LRAP ↑
		7.99±0.50	14.35±0.66	21.56±2.47	43.75±5.71	68.42±2.70	86.38±0.94	62.37±2.23
	✓	7.31±0.05	12.57±0.13	28.15±0.72	54.63±0.41	74.07±0.77	88.72±0.17	68.38±0.27
✓		7.35±0.10	13.04±0.22	25.61±1.03	53.03±0.76	68.90±0.40	87.13±0.24	67.53±0.56
✓	✓	**7.29±0.07**	**12.54±0.19**	**28.39±0.77**	**54.69±0.55**	**74.24±0.72**	**88.76±0.10**	**68.46±0.45**
PCG $\mathcal{L}_{align}$	PCG $\mathcal{L}_{match}$	HL ↓	RL ↓	Ma-AP ↑	Mi-AP ↑	Ma-AUC ↑	Mi-AUC ↑	LRAP ↑
		13.13±0.07	27.50±0.25	16.92±0.92	26.29±1.12	55.51±1.44	70.04±0.53	47.18±0.25
	✓	12.76±0.08	25.45±0.37	23.65±0.34	34.08±0.40	64.38±0.28	74.01±0.17	51.22±0.48
✓		13.02±0.19	27.69±0.26	18.20±1.73	28.12±2.43	56.62±1.78	70.33±0.36	48.02±1.21
✓	✓	**12.68±0.05**	**25.37±0.30**	**25.31±0.34**	**34.62±0.08**	**65.29±0.24**	**74.32±0.19**	**51.38±0.38**

4.2 Ablation Study

To evaluate the contributions of the LFR and the GCC, we conduct an ablation study to evaluate the model performance *w.r.t.* the corresponding losses, *i.e.*, $\mathcal{L}_{align}$ and $\mathcal{L}_{match}$. As shown in Table 3, the variant without both losses yields the weakest performance, confirming the functions of the proposed modules. Individually, $\mathcal{L}_{align}$ effectively reduces classification errors by separating entangled features and clearing decision boundaries, while $\mathcal{L}_{match}$ boosts ranking metrics (*e.g.*, Mi-AP) by mitigating optimization inconsistency induced by heterogeneous neighbors. Ultimately, the complete DF^{2}Net achieves the best performance across all metrics, demonstrating a strong synergistic effect where LFR and GCC mutually reinforce each other to facilitate the fusion process.

4.3 Further Analysis

More experiments on hyperparameter sensitivity, convergence analysis, and complexity analysis are provided in **Appendix** B.2, B.3, and B.4, respectively.

5 Conclusion

In this paper, we propose DF^{2}Net to address the complex challenges of the MLNC task, which models homogeneous and heterogeneous contexts from a frequency perspective. To jointly mine the beneficial complementary information from both homogeneous and heterogeneous contexts, DF^{2}Net decouples topological information into these two frequency signals to capture their complementary semantics simultaneously. To effectively integrate these complementary signals, we introduce the JROC module. By synergizing the LFR and GCC strategies, JROC successfully mitigates feature entanglement, aligns the optimization consistency among neighbors, and facilitates the dual-frequency fusion process. Extensive experiments across five graph datasets validate the superiority of DF^{2}Net over existing state-of-the-art methods.

AI USE STATEMENT

In this work, we used generative AI tools for assisting in the writing polishment. We have not used generative AI tools for generating synthetic datasets, and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools for identifying relevant literature. We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey, and LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

A link to an anonymous downloadable source code is provided supplementary materials. For theoretical results, clear explanations of any assumptions and a complete proof of the claims can be included in the Appendix. For any datasets used in the experiments, complete descriptions of the datasets are provided in the Appendix.

REFERENCES

Uchenna Akujuobi, Han Yufei, Qiannan Zhang, and Xiangliang Zhang. Collaborative graph walk for semi-supervised multi-label node classification. In *IEEE International Conference on Data Mining*, pp. 1–10, 2019.

Yuanchen Bei, Weizhi Chen, Hao Chen, Sheng Zhou, Carl Yang, Jiapei Fan, Longtao Huang, and Jiajun Bu. Correlation-aware graph convolutional networks for multi-label node classification. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 37–48, 2025.

Hao Chen, Yuanchen Bei, Qijie Shen, Yue Xu, Sheng Zhou, Wenbing Huang, Feiran Huang, Senzhang Wang, and Xiao Huang. Macro graph neural networks for online billion-scale recommender systems. In *Proceedings of the ACM Web Conference*, pp. 3598–3608, 2024a.

Zhaoliang Chen, Zhihao Wu, Zhenghong Lin, Shiping Wang, Claudia Plant, and Wenzhong Guo. Agnn: Alternating graph-regularized neural networks to alleviate over-smoothing. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13764–13776, 2024b.

Zhaoliang Chen, Zhihao Wu, Ylli Sadikaj, Claudia Plant, Hong-Ning Dai, Shiping Wang, Yiu-Ming Cheung, and Wenzhong Guo. Adedgedrop: Adversarial edge dropping for robust graph neural networks. *IEEE Transactions on Knowledge and Data Engineering*, pp. 4948–4961, 2025.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006.

Liang Duan, Xiang Chen, Wenjie Liu, Daliang Liu, Kun Yue, and Angsheng Li. Structural entropy based graph structure learning for node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8372–8379, 2024.

Kaisheng Gao, Jing Zhang, and Cangqi Zhou. Semi-supervised graph embedding for multi-label graph node classification. In *International Conference on Web Information Systems Engineering*, pp. 555–567, 2019.

Ziqi Gao, Chenran Jiang, Jiawen Zhang, Xiaosen Jiang, Lanqing Li, Peilin Zhao, Huanming Yang, Yong Huang, and Jia Li. Hierarchical graph learning for protein–protein interaction. *Nature Communications*, 14(1):1093, 2023.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, pp. 22118–22133, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, pp. 1–14, 2017.

Yuena Lin, Haichun Cai, Yi Shan, Hao Wei, Yongjian Deng, Zhen Yang, and Gengyu Lyu. Dfˆ 2-vb: Dual-level fuzzy fusion with view-specific boosting for multi-view multi-label classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33859–33868, 2026.

Chengliang Liu, Jie Wen, Yong Xu, Bob Zhang, Liqiang Nie, and Min Zhang. Reliable representation learning for incomplete multi-view missing multi-label classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4940–4956, 2025.

Weigang Lu, Ziyu Guan, Wei Zhao, Yaming Yang, Yibing Zhan, Yiheng Lu, and Dapeng Tao. Agmixup: Adaptive graph mixup for semi-supervised node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19143–19151, 2025.

Gengyu Lyu, Xiang Deng, Yanan Wu, and Songhe Feng. Beyond shared subspace: A view-specific fusion for multi-view multi-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7647–7654, 2022.

Gengyu Lyu, Zhen Yang, Xiang Deng, and Songhe Feng. L-vsm: Label-driven view-specific fusion for multiview multilabel classification. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6569–6583, 2025.

Yi Shan, Liyang Gao, Yuena Lin, Zhen Yang, Gengyu Lyu, and Honggui Han. Hypergraph-based multi-view multi-label classification via adaptive high-order semantic fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 25269–25277, 2026.

Min Shi, Yufei Tang, Xingquan Zhu, and Jianxun Liu. Multi-label graph convolutional network representation learning. *IEEE Transactions on Big Data*, 8(5):1169–1181, 2022.

Zixing Song, Ziqiao Meng, Yifei Zhang, and Irwin King. Semi-supervised multi-label learning for graph-structured data. In *Proceedings of the ACM International Conference on Information & Knowledge Management*, pp. 1723–1733, 2021.

Yifei Sun, Zemin Liu, Bryan Hooi, Yang Yang, Rizal Fathony, Jia Chen, and Bingsheng He. Multi-label node classification with label influence propagation. In *International Conference on Learning Representations*, 2025.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

Yanqiang Tu, Gengyu Lyu, Dian Zhang, Xiaozhu Jing, Wei Ke, and Wuman Luo. Robust multi-view multi-label learning via uncertainty-gated diffusion and dynamic graph mrmr. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4683–4694, 2026.

Hao Wei, Yongjian Deng, Qiuru Hai, Yuena Lin, Zhen Yang, and Gengyu Lyu. Multi-view multi-label classification via view-label matching selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 21456–21464, 2025.

Jie Wen, Jiang Long, Xiaohuan Lu, Chengliang Liu, Xiaozhao Fang, and Yong Xu. Partial multiview incomplete multilabel learning via uncertainty-driven reliable dynamic fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(1):236–250, 2026.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International Conference on Machine Learning*, pp. 6861–6871, 2019.

Yanan Wu, He Liu, Songhe Feng, Yi Jin, Gengyu Lyu, and Zizhang Wu. Gm-mlic: Graph matching based multi-label image classification. In *International Joint Conference on Artificial Intelligence*, pp. 1179–1185, 2021.

Lin Xiao, Pengyu Xu, Liping Jing, Uchenna Akujuobi, and Xiangliang Zhang. Semantic guide for semi-supervised few-shot multi-label node classification. *Information Sciences*, 591:235–250, 2022.

Bin Yu, Hengjie Xie, Mingjie Cai, and Weiping Ding. Mg-gcn: Multi-granularity graph convolutional neural network for multi-label classification in multi-label information system. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(1):288–299, 2024.

Luying Zhong, Renjie Lin, Jiayin Li, Shiping Wang, and Zheyi Chen. Bridging and compressing feature and semantic spaces for robust graph neural networks: An information theory perspective. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4571–4582, 2024.

Luying Zhong, Zhaoliang Chen, Zhihao Wu, Shide Du, Zheyi Chen, and Shiping Wang. Learnable graph convolutional network with semisupervised graph information bottleneck. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):433–446, 2025.

Cangqi Zhou, Hui Chen, Jing Zhang, Qianmu Li, Dianming Hu, and Victor S Sheng. Multi-label graph node classification with label attentive neighborhood convolution. *Expert Systems with Applications*, 180:115063, 2021.

Yonghua Zhu, Lei Feng, Zhenyun Deng, Yang Chen, Robert Amor, and Michael Witbrock. Robust node classification on graph data with graph and label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17220–17227, 2024.

A THEOREMS AND PROOFS

Theorem 3.1 (Low-Frequency Property of GCN) *The neighborhood aggregation in GCN acts as a low-pass filter, which inherently preserves low-frequency signals.*

Proof. From the perspective of graph signal processing (GSP), the GCN updates node representations via neighborhood aggregation, formulated as $\hat{\boldsymbol{A}}\boldsymbol{X}$. Using the definition of the normalized graph Laplacian $\tilde{\boldsymbol{L}} = \boldsymbol{I} - \hat{\boldsymbol{A}}$, we can express the aggregation matrix as $\hat{\boldsymbol{A}} = \boldsymbol{I} - \tilde{\boldsymbol{L}}$. Performing symmetric eigendecomposition yields $\tilde{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{U}$ is the eigenvector matrix serving as the graph Fourier basis, and $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues. In GSP, these eigenvalues denote the graph frequencies, which typically lie in the range $\lambda \in [0, 2]$. A smaller eigenvalue (low frequency, $\lambda \to 0$) indicates that connected nodes share similar features, reflecting homogeneous interactions. Conversely, a larger eigenvalue (high frequency, $\lambda \to 2$) indicates sharp variations across edges, reflecting heterogeneous interactions.

In the spectral domain, applying the $\hat{\boldsymbol{A}}$ to the graph signal $\boldsymbol{X}$ is mathematically equivalent to modulating the signals with a spectral filter function $h(\lambda) = 1 - \lambda$. We analyze the absolute filter response $|h(\lambda)|$ alongside the spectral effect to explain that GCN acts as a low-pass filter.

GCN introduces self-loops to ensure nodes retain their own features during aggregation. From a spectral perspective, this renormalization mechanism compresses the eigenvalue spectrum of the normalized Laplacian. While the original eigenvalues λ can reach up to 2, the addition of self-loops strictly bounds the maximum eigenvalue λ_{max} to be smaller (Wu et al., 2019). Consequently, the absolute filter response $|h(\lambda)|$ for all non-zero frequencies is strictly restricted to be less than 1, *i.e.*, $|1 - \lambda| < 1$ for $\lambda \in (0, \lambda_{max}]$.

When stacking K GCN layers, this filtering effect is exponentially amplified as $h_K(\lambda) = (1 - \lambda)^K$. For low-frequency signals ($\lambda \to 0$), the response approaches 1, namely $h(\lambda) \approx 1$, which means that the filter tends to preserve low-frequency signals. For high-frequency signals, since $|1 - \lambda| < 1$, their amplitudes suffer from exponential decay and rapidly vanish as K increases. Therefore, the GCN intrinsically coupled with the self-loop mechanism, operates as a low-pass filter.

Theorem 3.2 (High-Frequency Property of Discrepancy) *The discrepancy between an anchor node and its neighborhoods acts as a high-pass filter and captures high-frequency signals.*

Proof. The difference between a node intrinsic representation and its aggregated neighborhood representation is formulated as $\boldsymbol{H} - \hat{\boldsymbol{A}}\boldsymbol{H}$, which translates to the normalized Laplacian operation $(\boldsymbol{I} - \hat{\boldsymbol{A}})\boldsymbol{H} = \tilde{\boldsymbol{L}}\boldsymbol{H}$. Through spectral decomposition $\tilde{\boldsymbol{L}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, the graph spectral filtering paradigm is defined as $\boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^\top\boldsymbol{H}$. This transformation indicates that the spectral filter function strictly matches the eigenvalue itself, namely $h(\lambda) = \lambda$. Therefore, it forcefully suppresses low-frequency signals to prevent the absolute filter response $|h(\lambda)|$ from approaching zero, while preserving and amplifying high-frequency signals. Consequently, this difference intrinsically functions as a high-pass filter.

Theorem 3.3 *Let $\mathcal{L}_{aux}$ be the classification loss for training the auxiliary classifier, then the prediction error $\boldsymbol{g}_v = P_{aux}(v) - \boldsymbol{Y}_v$ is equivalent to the gradient of $\mathcal{L}_{aux}$ w.r.t. the logits.*

Proof. Let $\boldsymbol{z}_v \in \mathbb{R}^L$ denote the logits of the auxiliary classifier for node v, where L is the total number of labels. The prediction is obtained via the Sigmoid function:

$$P_{aux}(v) = \sigma(\boldsymbol{z}_v) = \frac{1}{1 + \exp(-\boldsymbol{z}_v)}. \tag{15}$$

For node v and its j-th label, the prediction probability is $\boldsymbol{p}_{v,j} = \sigma(\boldsymbol{z}_{v,j})$, and the ground truth is $\boldsymbol{y}_{v,j} \in \{0, 1\}$. The classification loss ℓ_{aux} for node v is formulated as:

$$\ell_{aux} = -\sum_{j=1}^{L} \left[\boldsymbol{y}_{v,j} \log(\boldsymbol{p}_{v,j}) + (1 - \boldsymbol{y}_{v,j}) \log(1 - \boldsymbol{p}_{v,j})\right]. \tag{16}$$

According to the chain rule, the gradient of the loss ℓ_{aux} *w.r.t.* the logit $\boldsymbol{z}_{v,j}$ is computed as:

$$\frac{\partial \ell_{aux}}{\partial \boldsymbol{z}_{v,j}} = \frac{\partial \ell_{aux}}{\partial \boldsymbol{p}_{v,j}} \frac{\partial \boldsymbol{p}_{v,j}}{\partial \boldsymbol{z}_{v,j}}. \tag{17}$$

Table 4: Statistics of the adopted public datasets.

Datasets	Humloc	PCG	PPI	OGB-Proteins	Delve
Nodes	3,106	3,233	14,755	132,534	1,229,280
Edges	18,496	37,351	225,270	39,561,252	4,322,275
Features	32	32	50	8	300
Classes	14	15	121	112	20
Domain	PPI	PPI	PPI	PPI	Wikipedia Network
Node Type	Protein	Protein	Protein	Protein	Paper
Label Type	Subcellular Location	Phenotype	Gene Ontology Category	Functions	Topics

First, we compute the derivative of the loss *w.r.t.* $\boldsymbol{p}_{v,j}$:

$$\frac{\partial \ell_{aux}}{\partial \boldsymbol{p}_{v,j}} = -\frac{\boldsymbol{y}_{v,j}}{\boldsymbol{p}_{v,j}} + \frac{1-\boldsymbol{y}_{v,j}}{1-\boldsymbol{p}_{v,j}} = \frac{\boldsymbol{p}_{v,j}-\boldsymbol{y}_{v,j}}{\boldsymbol{p}_{v,j}(1-\boldsymbol{p}_{v,j})}. \tag{18}$$

Second, the derivative of the Sigmoid function $\boldsymbol{p}_{v,j}$ *w.r.t.* $\boldsymbol{z}_{v,j}$ is:

$$\frac{\partial \boldsymbol{p}_{v,j}}{\partial \boldsymbol{z}_{v,j}} = \boldsymbol{p}_{v,j}(1-\boldsymbol{p}_{v,j}). \tag{19}$$

Multiplying these two partial derivatives together, we have:

$$\frac{\partial \ell_{aux}}{\partial \boldsymbol{z}_{v,j}} = \frac{\boldsymbol{p}_{v,j}-\boldsymbol{y}_{v,j}}{\boldsymbol{p}_{v,j}(1-\boldsymbol{p}_{v,j})} \cdot \boldsymbol{p}_{v,j}(1-\boldsymbol{p}_{v,j}) = \boldsymbol{p}_{v,j}-\boldsymbol{y}_{v,j}. \tag{20}$$

Generalizing this conclusion to the vector form for all labels, the gradient of the loss *w.r.t.* the logit vector $\boldsymbol{z}_v$ corresponds to the prediction error:

$$\nabla \ell_{aux} = P_{aux}(v) - \boldsymbol{Y}_v = \boldsymbol{g}_v. \tag{21}$$

Therefore, the prediction error $\boldsymbol{g}_v$ naturally serves as the gradient.

B Experimental Settings

B.1 Datasets and Settings

We comprehensively evaluate DF^{2}Net on five widely-used public multi-label graph datasets, with detailed statistics summarized in Table 4:

Humloc is a biological network where nodes correspond to proteins and edges denote their physical or functional interactions. The labels indicate the subcellular locations of proteins, effectively capturing the biological phenomenon that a single protein may simultaneously reside in multiple cellular compartments.

PCG is another bioinformatics graph representing complex genetic interactions; its nodes stand for proteins or genes, while edges capture their interaction mechanisms. The multi-label annotations align with the diverse biological functions and pathways in which these entities jointly participate.

PPI comprises protein-protein interaction networks extracted from various human tissues. In this dataset, nodes represent human proteins interconnected by edges signifying biological linkages, and the labels describe multiple concurrent cellular functions or biological states, making it highly suitable for evaluating complex node-label correlations.

Ogbn-Proteins is a large-scale and challenging graph dataset introduced by the Open Graph Benchmark (Hu et al., 2020). It models proteins from eight different species as nodes, with edges representing biologically meaningful associations (e.g., co-expression or homology). The prediction task involves assigning multiple protein functions, requiring the model to effectively handle massive structural scales and inter-species heterogeneity.

Delve is a large-scale web document graph where nodes are text documents and edges represent hyperlink connections. The labels denote semantic categories or topics, reflecting the reality that a single document inherently covers multiple overlapping thematic subjects.

Table 5: Hyperparameter settings for different datasets.

Datasets	Humloc	PCG	PPI	OGB-Proteins	Delve
τ	0.1	1.0	0.05	0.07	0.05
λ_1	3e-1	1e-2	1e-1	5e-2	5e-3
λ_2	1.5e-1	1.5	5e-6	5e-4	5e-5
d	64	64	64	64	64
B	—	—	4096	10240	10240
Dropout Ratio	0.4	0.3	0	0.1	0
Learning Rate	3e-3	5e-3	5e-3	5e-3	5e-3
Layer Number	2	2	2	2	2

All experiments are implemented in PyTorch and conducted on a server with NVIDIA GeForce 3090. The hyperparameter settings of DF^{2}Net are listed in Table 5. τ is the temperature coefficient, λ_1 and λ_2 are two penalty coefficients, d is the output dimension of the encoder, and Layer Number is the number of GNN layers. B is the batch size, where "—" means that the whole multi-label graph is input into the model.

B.2 Hyperparameter Sensitivity

We conduct a hyperparameter sensitivity analysis *w.r.t.* the temperature coefficient τ, and the penalty coefficients λ_1 and λ_2, to analyze their effects. As illustrated in Figure 4, three hyperparameters affect the model performance, which generally exhibits a consistent "increase-then-decrease" trend across four datasets. Furthermore, the sensitivity to these hyperparameters varies depending on the scale of the datasets. On smaller datasets such as Humloc and PCG, the performance exhibit more clear fluctuations. In contrast, on larger-scale dataset like Delve, the performance curves remain more stable, since large datasets provide sufficient structure and feature information for nodes, rendering the model less sensitive to hyperparameter variations.

B.3 Convergence Analysis

Figure 5 illustrates the convergence trend of DF^{2}Net during training. Across all subplots, the training loss consistently exhibits a rapid and steep decline during the initial training epochs. This indicates that the model can efficiently optimize the joint objective function. Specifically, the loss curves for the Humloc and Delve datasets are smooth, quickly reaching a steady plateau and demonstrating stable optimization. Collectively, these consistent trends empirically validate that the joint representation-optimization strategy in DF^{2}Net can converge across different datasets.

B.4 Complexity Analysis

In this section, we analyze the complexity of the proposed DF^{2}Net under the mini-batch setting. Let B be the number of sampled nodes in a batch, E_B be the number of sampled edges, d be the representation dimension, and L be the total number of labels. The overall complexity mainly stems from three components: (1) Dual-frequency representation extraction requires $\mathcal{O}(E_B d + Bd^2)$ for neighborhood aggregation and MLP feature transformations. (2) Label-aware frequency fusion takes $\mathcal{O}(Bd^2 + BdL)$ for generating state representations and the fusion process. (3) The JROC module requires $\mathcal{O}(BdL + E_B L)$ to compute the auxiliary predictions, calculate the gradient consistency similarities across edges, and perform node-label alignment. Summing these operations, the overall complexity per mini-batch is bounded by $\mathcal{O}(E_B(d + L) + B(2d^2 + dL))$.

C Algorithm

We provide an overview of the training process in Algorithm 1.

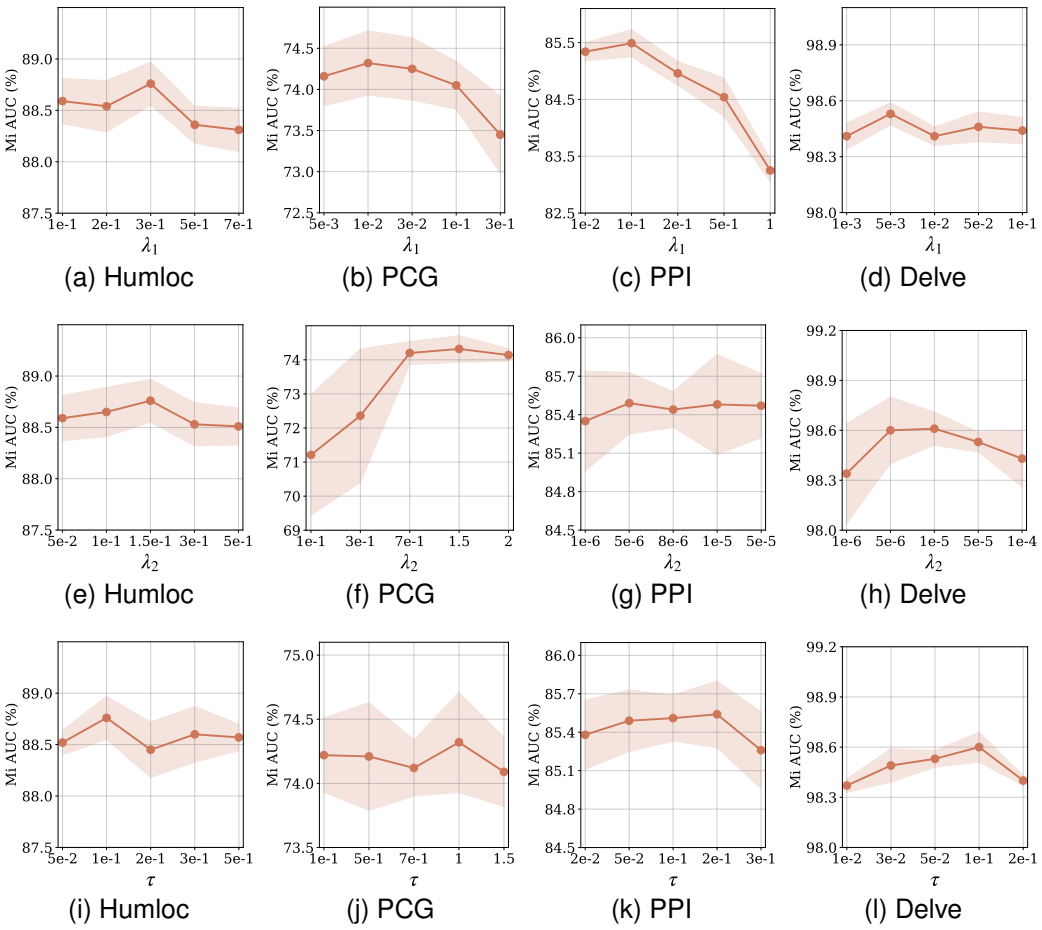


Figure 4: Hyperparameter sensitivity analysis *w.r.t.* the temperature coeffcient τ, and the penalty coefficients λ_1 and λ_2.

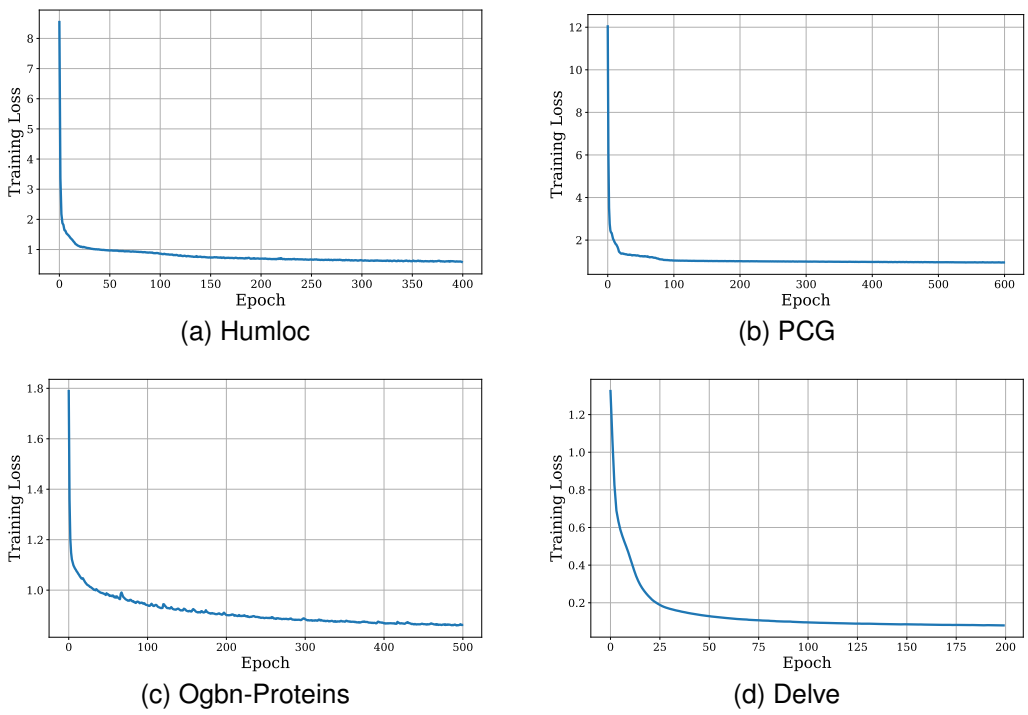


Figure 5: The convergence analysis of DF^{2}Net.

Algorithm 1 Training Process of DF^{2}Net

Input: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with adjacency matrix $\boldsymbol{A}$, Node feature matrix $\boldsymbol{X}$, Label matrix of training samples $\tilde{\boldsymbol{Y}}$, Number of epochs T, Learning rate η, Hyperparameters λ_1, λ_2, τ.
Output: Network parameter Θ and label representation $\boldsymbol{E}$.

Initialize model parameters Θ and $\boldsymbol{E}$;
for epoch $= 1$ to T **do**
  // 1. Dual-Frequency Representation Extraction
  $\boldsymbol{H}_{low} = \text{GCN}(\boldsymbol{A}, \boldsymbol{X})$, $\boldsymbol{H}_{self} = \text{MLP}(\boldsymbol{X})$;
  $\boldsymbol{H}_{high} = \boldsymbol{H}_{self} - \boldsymbol{H}_{low}$;
  // 2. Label-Aware Frequency Fusion
  Calculate structure state representation: $\boldsymbol{H}_{state} = \text{MLP}([\boldsymbol{H}_{low} \parallel \boldsymbol{H}_{high}])$;
  Compute adaptive fusion weight: $\boldsymbol{\alpha} = \sigma(\boldsymbol{H}_{state}\boldsymbol{E}^\top)$;
  Predictions: $\boldsymbol{Z}_{low} = \boldsymbol{H}_{low}\boldsymbol{W}$, $\boldsymbol{Z}_{self} = \boldsymbol{H}_{self}\boldsymbol{W}$;
  Final prediction: $\boldsymbol{Z}_{final} = \boldsymbol{\alpha} \odot \boldsymbol{Z}_{\text{low}} + (\boldsymbol{1} - \boldsymbol{\alpha}) \odot \boldsymbol{Z}_{self}$;
  Calculate the loss $\mathcal{L}_{freq} = \text{BCE}(\sigma(\boldsymbol{Z}_{final}), \tilde{\boldsymbol{Y}})$;
  // 3. Label-guided Feature Refinement
  Calculate the alignment loss $\mathcal{L}_{align} = (\boldsymbol{H}_{state}, \boldsymbol{E}, \tilde{\boldsymbol{Y}}, \tau)$;
  // 4. Gradient Consistency Calibration
  Introduce an auxiliary classifier to generate the prediction P_{aux} and gradient $\boldsymbol{g}_v$ with $\mathcal{L}_{aux}$;
  Compute aggregated neighbor gradient $\boldsymbol{g}_{\mathcal{N}(v)}$;
  Calculate the consistency: $\boldsymbol{S}_{sup}(v) = \frac{\boldsymbol{g}_v \odot \boldsymbol{g}_{\mathcal{N}(v)}}{|\boldsymbol{g}_v||\boldsymbol{g}_{\mathcal{N}(v)}|+\epsilon}$;
  Transform the interval of similarity: $\boldsymbol{S}(v) = \frac{\boldsymbol{S}_{sup}(v)+\boldsymbol{1}}{2}$;
  Calculate the matching loss: $\mathcal{L}_{match} = \text{MSE}(\boldsymbol{\alpha}, \boldsymbol{S})$;
  // 5. Optimization
  Calculate the total loss: $\mathcal{L} = \mathcal{L}_{freq} + \mathcal{L}_{aux} + \lambda_1\mathcal{L}_{align} + \lambda_2\mathcal{L}_{match}$;
  Backpropagate $\mathcal{L}$ and update Θ, $\boldsymbol{E}$;
end for
return Θ, $\boldsymbol{E}$