ES-GNN: Generalizing Graph Neural Networks Beyond Homophily with Edge Splitting

Jingwei Guo, Kaizhu Huang*, Rui Zhang, and Xinping Yi

Abstract—While Graph Neural Networks (GNNs) have achieved enormous success in multiple graph analytical tasks, modern variants mostly rely on the strong inductive bias of homophily. However, real-world networks typically exhibit both homophilic and heterophilic linking patterns, wherein adjacent nodes may share dissimilar attributes and distinct labels. Therefore, GNNs smoothing node proximity holistically may aggregate both task-relevant and irrelevant (even harmful) information, limiting their ability to generalize to heterophilic graphs and potentially causing non-robustness. In this work, we propose a novel Edge Splitting GNN (ES-GNN) framework to adaptively distinguish between graph edges either relevant or irrelevant to learning tasks. This essentially transfers the original graph into two subgraphs with the same node set but complementary edge sets dynamically. Given that, information propagation separately on these subgraphs and edge splitting are alternatively conducted, thus disentangling the task-relevant and irrelevant features. Theoretically, we show that our ES-GNN can be regarded as a solution to a *disentangled graph denoising problem*, which further illustrates our motivations and interprets the improved generalization beyond homophily. Extensive experiments over 11 benchmark and 1 synthetic datasets not only demonstrate the effective performance of ES-GNN but also highlight its robustness to adversarial graphs and mitigation of the over-smoothing problem.

Index Terms—Graph Neural Networks, Heterophilic Graphs, Disentangled Representation Learning, Graph Mining.

1 Introduction

As a ubiquitous data structure, graph can symbolize complex relationships between entities in different domains. For example, knowledge graphs describe the interconnections between real-world events, and social networks store the online interactions between users. With the flourishing of deep learning models on graph-structured data, graph neural networks (GNNs) emerge as one of the most powerful techniques in recent years. Owing to their remarkable performance, GNNs have been widely adopted in multiple graph-based learning tasks, such as link prediction, node classification, and recommendation [1], [2], [3], [4].

Modern GNNs are mainly built upon a message passing framework [5], where nodes' representations are learned by aggregating their transformed neighbors iteratively. From the graph signal denoising viewpoint, this mechanism could be seen as a low-pass filter [6], [7], [8], [9] that smooths the signals between adjacent nodes. Several works [8], [10], [11], [12], [13], [14], [15] refer this to smoothness or homophily assumption in GNNs. Notably, they work well on homophilic (assortative) graphs, from which the proximity information of nodes can be utilized to predict their labels [16]. However, real-world networks are typically abstracted from complex systems, and sometimes display heterophilic (disassortative) properties whereby the opposite objects are attracted to each other [17]. For instance, different types of amino acids are mostly interacted in many protein structures [10], and most people in heterosexual dating networks prefer to link with others of the opposite gender. Recent studies [10], [11], [12], [13], [14], [18], [19], [20], [21], [22], [23] have shown that the conventional neighborhood aggregation strategy may not only cause the over-smoothing problem [24], [25] but also severely hinder the generalization performance of GNNs beyond homophily.

One reason why current GNNs perform poorly on heterophilic graphs, could be the mismatch between the labeling rules of nodes and their linking mechanism. The former is the target that GNNs are expected to learn for classification tasks, while the latter specifies how messages pass among nodes for attaining this goal. In homophilic scenarios, both of them are similar in the sense that most nodes are linked because of their commonality which therefore leads to identical labels. In heterophilic scenarios, however, the motivation underlying why two nodes get connected may be ambiguous to the classification task. Let us take the social network within a university as an example, where students from different clubs can be linked usually due to taking the same classes and/or being roommates but not sharing the same hobbies. Namely, the task-relevant and irrelevant (or even harmful) information is typically mixed into node neighborhood under heterophily. However, current methods usually fail to recognize and differentiate these two types of information within nodes' proximity, as illustrated in Fig. 1. As a consequence, the learned representations are prone to be entangled with false information, leading to non-robustness and sub-optimal performance.

Once the issue of GNNs' learning beyond homophily is identified, a natural question arises: *Can we design a new type of GNNs that is adaptive to both homophilic and heterophilic scenarios?* Well formed designs should be able to

- *J. Guo is with University of Liverpool, Liverpool, UK*
  E-mail: jingwei.guo@liverpool.ac.uk
- *K. Huang is with Duke Kunshan University, Suzhou, China*
  E-mail: kaizhu.huang@dukekunshan.edu.cn
- *R. Zhang is with Xi'an Jiaotong-Liverpool University, Suzhou, China*
  E-mail: rui.zhang02@xjtlu.edu.cn
- *X. Yi is with Southeast University, Nanjing, China*
  E-mail: xyi@seu.edu.cn

**Corresponding author: Kaizhu Huang*

identify the node connections irrelevant to learning tasks, and substantially extract the most correlated information for prediction. However, the assortativity of real-world networks is usually agnostic. Even worse, the features of nodes are typically full of noises, where similarity or dissimilarity between connected ones may not actually reflect their class relations. Existing techniques including [18], [26], [27] usually parameterize graph edges with node similarity or dissimilarity, and cannot well assess the correlation between node connections and the downstream target.

In this paper, we propose ES-GNN, an end-to-end graph learning framework that generalizes GNNs on graphs with either homophily or heterophily. Without loss of generality, we make an assumption that two nodes get connected mainly because they share some similar features, which are however unnecessarily just relevant to the learning task. In other words, nodes may be linked due to similar features, either relevant or irrelevant to the task. This implicitly divides the original graph edges into two complementary sets, each of which represents a latent relation between nodes. Thanks to the proximity smoothness, aggregating node features individually on each edge set should disentangle the task-relevant and irrelevant features. Meanwhile, these disentangled representations potentially reflect node similarity in two aspects (task-relevant and irrelevant). As such, they can be better utilized to split the original graph edges more precisely. Motivated by this, the proposed framework integrates GNNs with an interpretable edge splitting (ES), to jointly partition network topology and disentangle node features.

Technically, we design a residual scoring mechanism, executed within each ES-layer, to distinguish the task-relevant and irrelevant graph edges. The node features are then aggregated separately on these connections to produce disentangled representations, based on which graph edges can be classified more accurately in the next ES-layer. Finally, the task-relevant representations are granted for prediction. Meanwhile, an Irrelevant Consistency Regularization (ICR) is developed to regulate the task-irrelevant representations with the potential label-disagreement between adjacent nodes, for further reducing the classification-harmful information from the final predictive target. To interpret our new algorithm theoretically, generalizing the *standard smoothness assumption* [8], we also conduct some analysis on ES-GNN and establish its connection with a *disentangled graph signal denoising problem*. In summary, the main contributions of this work are four-fold:

- We propose a novel framework called ES-GNN for node classification tasks with one plausible hypothesis, which enables GNNs to go beyond the strong homophily assumption on graphs.
- We theoretically prove that our ES-GNN is equivalent to solving a graph denoising problem with a *disentangled smoothness assumption*, which interprets its good performance on different types of networks.
- Extensive evaluations across 11 benchmark and 1 synthetic datasets illustrate ES-GNN's efficacy on graphs with varying homophily levels, achieving an average error reduction of 5.8% over a broad spectrum of competitive methods.
- Importantly, ES-GNN is able to alleviate the over-smoothing problem and enjoys remarkable robustness against adversarial graphs. This shows that ES-GNN could still lead to excellent performance even if the *disentangled smoothness assumption* may not hold practically.

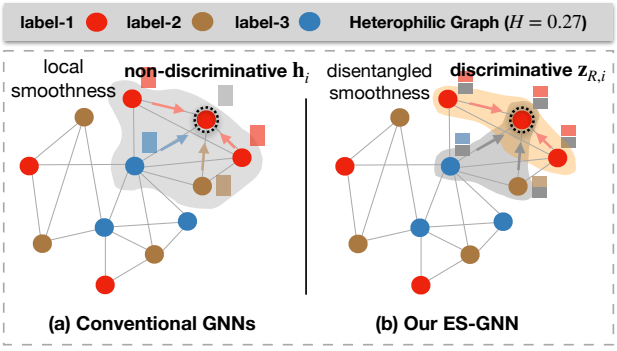


Fig. 1. A toy example to show differences between conventional GNNs and our ES-GNN in aggregating node features. Conventional GNNs with local smoothness tend to produce non-discriminative representations on heterophilic graphs, while our ES-GNN is able to disentangle and exclude the task-harmful features from the final predictive target.

2 Preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with node set $\mathcal{V} = \{v_n\}_{n=1}^{N}$ and edge set $\mathcal{E}$, where $N = |\mathcal{V}|$ refers to node number and $(v_i, v_j) \in \mathcal{E}$ if two distinct nodes v_i, v_j are connected. We use $\mathcal{N}_i$ to denote the 1-hop neighborhood of node v_i and define the adjacency matrix as $\mathbf{A} \in \mathbb{R}^{N\times N}$ where $\mathbf{A}_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and 0 otherwise. The degree matrix $\mathbf{D}$ can be obtained by summing the row of $\mathbf{A}$ into a diagonal matrix. As our ES-GNN disentangles the original graph into the task-relevant and irrelevant subgraphs, we will denote their adjacency matrixes respectively as $\mathbf{A}_{\mathrm{R}}$ and $\mathbf{A}_{\mathrm{IR}}$ in this paper. Nodes are usually associated with a feature matrix $\mathbf{X} \in \mathbb{R}^{N\times F}$ where F refers to the number of raw feature and $\mathbf{X}_{[i,:]}$ is the i-th row of $\mathbf{X}$ pertinent to node v_i. For node classification tasks, each node is assigned with a label c_i out of $C \leq N$ classes and have a ground truth one-hot vector $\mathbf{y}_i \in \mathbb{R}^C$. In this context, real-world graphs can be divided into homophilic and heterophilic ones based on the extent of similarity (or dissimilarity) in class labels among connected nodes. To quantify this level of homophily, researchers have developed various metrics. Among these, edge homophily $\mathcal{H}$ is a widely used metric that calculates the proportion of edges connecting nodes with identical labels, expressed as: $\mathcal{H} = |\{(v_i, v_j)|\mathbf{y}_i = \mathbf{y}_j, (v_i, v_j) \in \mathcal{E}\}|/|\mathcal{E}|$. This metric ranges from 0 (high heterophily) to 1 (high homophily). Recently, more nuanced metrics such as class homophily $\mathcal{H}_{\text{class}}$ and adjusted homophily $\mathcal{H}_{\text{adjusted}}$ have been proposed in works [28] and [29], respectively. These metrics take into account potential class imbalance and the variability in class number across different datasets, offering a more accurate estimation.

3 Background and Related Work

In this section, we provide the necessary background and elucidate the connections between our work and previous studies in the field (see subsections 3.1.1, 3.1.2 and 3.2.1).

3.1 Graph Neural Networks

The central idea of most GNNs is to utilize nodes' proximity information for building their representations for tasks, based on which great effort has been made in developing different variants [6], [26], [30], [31], [32], [33], [34], [35], [36], [37], [38], and understanding the nature of GNNs [8], [9], [39], [40], [41], [42]. Several works have proved that GNNs essentially behave as a low pass filter that smooths information within node surrounding [6], [7], [16], [43]. In line with this view, [9] and [8] show that a number of GNN models, such as GCN [30] adopting first-order Chebyshev expansion for efficient graph convolution, SGC [6] removing non-linearity of GCN, and GAT [26] parameterizing graph edges with an attention mechanism, can be seen as different optimization solvers to a graph signal denoising problem with a *smoothness assumption* upon connected nodes. All these results indicate that most GNNs are designed with a strong homophily hypothesis on the observed graphs while largely overlooking the important setting of heterophily, where node features and labels vary unsmoothly on graphs.

3.1.1 Connection to GNNs Tailored for Heterophily

This subsections briefly introduce GNNs tailored for addressing graphs under heterophilic scenarios and emphasize the differences between their approaches and ours.

To extend GNNs on heterophilic graphs, several works leverage the long-range information beyond nodes' proximity. Geom-GCN [44] extends the standard message passing with geometric aggregation in latent space. H2GCN [10] directly models the higher order neighborhoods for capturing the homophily-dominant information. WRGAT [14] transforms the input graph into a multi-relational graph, for modeling structural information and enhancing the assortativity level. GEN [13] estimates a suitable graph for GNNs' learning with multi-order neighborhood information and Bayesian inference as guide. GloGNN++ [23] captures the global homophily beyond immediate neighborhoods by learning a signed matrix to assess correlations among nodes. Another line of work emphasizes the proper utilization of node neighbors. The most common works employ attention mechanism [26], [45], however, they are still imposing smoothness within nodes' neighborhood albeit on the important members only [7], [8], [9]. Compared to that, FAGCN [18] adaptively models both similarities and dissimilarities between adjacent nodes. GPR-GNN [11] introduces a universal polynomial graph filter, by associating different hop neighbors with learnable weights in both positive and negative signs, so as to extract both low- and high-frequency information. ACM-GCN [46] proposes a multi-channel filtering approach that adaptively exploit both low- and high-frequency neighborhood information for each node. GOAL [47] enhances the modeling of intra- and inter-class node relationships through a graph complementary learning that recovers missing low- and high-frequency information in the original network topology.

However, most of them overlook the motivations why two nodes get connected, nor do they associate them with learning tasks, which is analyzed as one of the keys to generalize GNNs beyond homophily in this paper. In contrast, ES-GNN distinguishes graph edges as either relevant or irrelevant to the task. Such information acts as a guide to disentangle and exclude classification-harmful information from the final predictive target, and thus boosts GNNs' performance under heterophily.

3.1.2 Connection to GNNs Considering Task-Relevance

Following the previous discussion, where we highlighted ES-GNN's distinctive approach of discerning task-relevant from irrelevant information amidst GNNs designed for heterophily, we now situate this idea within GNN research that focuses on task-relevance. While the notion of emphasizing task-relevant information is not new, this subsection is dedicated to clearly outlining how our work aligns with and diverges from existing methodologies in this realm.

The concept of prioritizing task-relevance in GNNs has been extensively explored across various domains, such as topological denoising [48], [49], [50], graph pooling [51], augmentations [52], [53], contrastive learning [54], and structure learning [55], [56], [57]. Although [58] does not explicitly define task-relevance or irrelevance, it subtly explains the reasons for node connections by employing relational topic modeling with hierarchical graph information. Given our focus on supervised classification tasks, the forthcoming discussion will be expressly centered around this area. This focus ensures a contextual analysis of ES-GNN within the established research landscape, highlighting its unique contributions to task-oriented GNN development.

One should note that there are essential distinctions between our method and the existing works of NeuralSparse [48] and GCN-LPA [49], despite sharing the common goal of learning task-relevant edges. NeuralSparse employs a sparsification mechanism that can be trained with task loss, while GCN-LPA utilizes the outcome of label propagation as a guide to learn edge weights. Although these methods, like ours, actively select task-relevant edges to facilitate the extraction of task-relevant information, they primarily focus on this aspect, neglecting the potential benefits of modeling the opposite, task-irrelevant aspect. In contrast, our ES-GNN diverges by implementing a disentangled learning paradigm that partitions network topology and decouples node features into task-relevant and irrelevant parts. This explicit modeling of task-irrelevant information allows ES-GNN to further reduce noise and enhance the extraction of features with stronger correlations with the task. In other words, our approach not only focuses on selecting task-relevant edges but also strategically minimizes the impact of irrelevant information, therefore excelling in complex settings like heterophilic graphs (see Table 4).

Additionally, while both of our work and DOTIN [51] explore the enhancement of GNNs by identifying task-irrelevant graph information, the approaches we take diverge in their application and focus. While DOTIN focuses on streamlining graph classification by dropping task-irrelevant nodes to boost efficiency and scalability, our ES-GNN targets node-level tasks, emphasizing the discernment of task-relevance in graph edges to generalize GNNs beyond homophily. Extending the core principles of ES-GNN to graph-level tasks presents a promising future direction.

3.2 Disentangled Representation Learning

Disentangled representation learning, aimed at disentangling the explanatory latent variables within observed data into distinct dimensions [59], [60], has garnered considerable attention, particularly in the field of computer vision [61], [62], [63], [64], [65]. In recent years, there has been a progressive expansion of disentangled representation learning into the graph domain, addressing a wide spectrum of tasks. These range from foundational classification [33], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76] and generation challenges [77], [78], [79], [80], [81], in both supervised and unsupervised settings, to downstream applications like trajectory prediction [82], overlapping community detection [83], recommendation systems [84], [85], [86], [87], [88], and graph neural architecture search [89]. Given our focus on foundational Graph Neural Network (GNN) models, the following discussion will concentrate on the subset of research that employs disentangled representation learning to enhance the capabilities of GNNs within the realm of fundamental tasks.

For instance, DisenGCN [66] introduces a neighborhood routing mechanism to iteratively partition node neighborhoods into distinct segments, paving the way for disentangled node-level information learning. Following this, IPGDN [67] and LGD-GCN [73] further enhance the model by promoting independence among disentangled factors and integrating global graph information, respectively. At the graph-level, FactorGCN [33] takes a novel approach by factorizing the original graph into multiple subgraphs, aiming to highlight various graph aspects. VEPM [71] subsequently extends this learning paradigm by developing an edge generative model that incorporates community information to partition edges. Distinct from the aforementioned works, DisGNN [72] focuses explicitly on disentangling graph edges. It employs three pretext tasks to guide the learning process, aiming to enhance GNN performance under heterophily settings – a goal that aligns closely with our work.

Shifting the focus to unsupervised learning approaches, DGCL [68] introduces a factor-wise discrimination objective in a contrastive learning manner to disentangle graph-level representations. Building upon this foundation, IDGCL [70] further enhances this approach by promoting the independence among the disentangled latent representations. Complementing these at the node level, DSSL [69] advances graph self-supervised learning by simulating a graph generative process through latent variable modeling of semantic structures. This process effectively decouples diverse neighborhood contexts, particularly benefiting the analysis of non-homophilous graphs. In the realm of graph generation, NED-VAE [77] stands out as a unsupervised approaches by automatically disentangling latent factors in both nodes and edges. SND-VAE [78] further advances this field as the first disentangled generative model tailored for spatial networks. It adeptly uncovers both independent and dependent latent factors of spatial and network domains.

3.2.1 Connections to Disentangled GNNs

This subsection explores the relationships between our ES-GNN model and established disentangled GNNs, such as FactorGCN [33], VEPM [71] and DisGNN [72], specifically within the context of supervised settings. Our emphasis on supervised node classification tasks guides the selection of these comparative models to highlight the unique contributions and distinctions of our approach.

First, we acknowledge that our work shares a foundational similarity with both FactorGCN and VEPM: the aim to decompose the original network topology into multiple subgraphs for disentangling node features. However, there are three main differences: **1)** unlike FactorGCN, which allows an edge to belong to multiple subgraphs, resulting in potential overlap, our ES-GNN adopts an edge-splitting strategy that adaptively divides the original network topology into two mutually complementary subgraphs, ensuring $\mathbf{A}_{\mathrm{R}} + \mathbf{A}_{\mathrm{IR}} = \mathbf{A}$. In this aspect, VEPM is somewhat similar to ours, also producing complementary subgraphs by normalizing edge weights with a softmax layer. **2)** FactorGCN merely interprets the decomposed subgraphs as different graph aspects without providing any concrete meanings, and the number of latent factors requires manual selection across different graphs. While VEPM attributes community characteristics to these subgraphs, it falls short in linking these characteristics directly to the task at hand, nor does this approach address the variability in the community number needed across graphs. In contrast, our model uniquely generates two interpretable, task-relevant and irrelevant topologies adaptable to any graph, offering more meaningful and application-specific insights. **3)** FactorGCN and VEPM integrate all disentangled components towards the final prediction, with VEPM even remixing the disentangled feature representations for prediction using a "representation composer". Diverging from them, our ES-GNN focuses on segregating task-relevant from task-irrelevant features, allowing for the exclusion of classification-harmful information in the predictive process. This distinction is particularly beneficial in heterophilic contexts, where task-irrelevant information could easily obscure the target prediction, as empirically validated in our experiments (see Table 4).

Second, as previously mentioned when introducing DisGNN, both our method and DisGNN aim to enhance GNN performance on heterophilic graphs through explicit edge disentanglement. However, unlike DisGNN, which relies on multiple heuristic-based pretext tasks to supervise the edge disentanglement process, our approach requires only the addition of an Irrelevance Consistency Regularization (ICR) loss alongside the main task loss. This ICR loss, systematic in nature, adheres strictly to our core model principle as outlined in Hypothesis 1. Moreover, similar to FactorGCN and VEPM, DisGNN does not prioritize task relevance when utilizing disentangled components for prediction. This approach risks retaining misleading information in heterophilic scenarios and potentially compromises model performance.

4 Framework: ES-GNN

In this section, we propose an end-to-end graph learning framework, ES-GNN, generalizing Graph Neural Networks (GNNs) to arbitrary graph-structured data with either homophilic or heterophilic properties. An overview of ES-

GNN is given in Fig. 2. The central idea is to integrate GNNs with an interpretable edge splitting (ES) layer that adaptively partitions the network topology as guide to disentangle the task-relevant and irrelevant node features.

4.1 Edge Splitting Layer

The goal of this layer is to infer the latent relations underlying adjacent nodes on the observed graph, and distinguish between graph edges which could be relevant or irrelevant to learning tasks. Given a simple graph with an adjacency matrix $\mathbf{A}$ and node feature matrix $\mathbf{X}$, an ES-layer splits the original graph edges into two complementary sets, and thereby produces two partial network topologies with adjacency matrices $\mathbf{A}_{\mathrm{R}}, \mathbf{A}_{\mathrm{IR}} \in \mathbb{R}^{N\times N}$ satisfying $\mathbf{A}_{\mathrm{R}} + \mathbf{A}_{\mathrm{IR}} = \mathbf{A}$. We would expect $\mathbf{A}_{\mathrm{R}}$ storing the most correlated graph edges to the classification task, of which the rest is excluded and disentangled in $\mathbf{A}_{\mathrm{IR}}$. Therefore, analyzing the correlation between node connections and learning tasks comes into the first step.

However, existing techniques [18], [26], [27] mainly parameterize graph edges with node similarity or dissimilarity, while failing to explicitly correlate them with the prediction target. Even worse, as the assortativity of real-world networks is usually agnostic and node features are typically full of noises, the captured similarity/dissimilarity may not truly reflect the label-agreement/disagreement between nearby nodes. Consequently, the harmful-similarity between pairwise nodes from different classes could be mistakenly preserved for prediction. To this end, we present one plausible hypothesis below, whereby the explicit correlation between node connections and learning tasks is established automatically.

Hypothesis 1. *Two nodes get connected in a graph mainly due to their similarity in some features, which could be either relevant or irrelevant (even harmful) to the learning task.*

This hypothesis is assumed without losing generality to both homophilic and heterophilic graphs. For a homophilic scenario, e.g., in citation networks, scientific papers tend to cite or be cited by others from the same area, and both of them usually possess the common keywords uniquely appearing in their topics. For a heterophilic scenario, students having different interests are likely be connected because of the same classes and/or dormitory they take and/or live in, but neither has direct relation to the clubs they have joined. This inspires us to classify graph edges by measuring the similarity between adjacent nodes in two different aspects, i.e., *a graph edge is more relevant to a classification task if connected nodes are more similar in their task-relevant features, or otherwise.* Our experimental analysis in Section 6.6 further provides evidences that even when our Hypothesis 1 may not hold, most adversarial edges (considered as the task-irrelevant ones) can still be recognized though neither types of node similarity exists.

It is worthy mentioning that our hypothesis is not in contradiction to the "opposites attract", which could be intuitively explained by linking due to different but matching attributes. We believe the inherent cause to connection even in "opposites attract" may still be certain commonalities. For example, in heterosexual dating networks, people of the opposite sex are most likely connected because of their similar life values. Although these similarities may be inappropriate (or even harmful) in distinguishing genders, modeling and disentangling them from the final predictive target might be still of great importance.

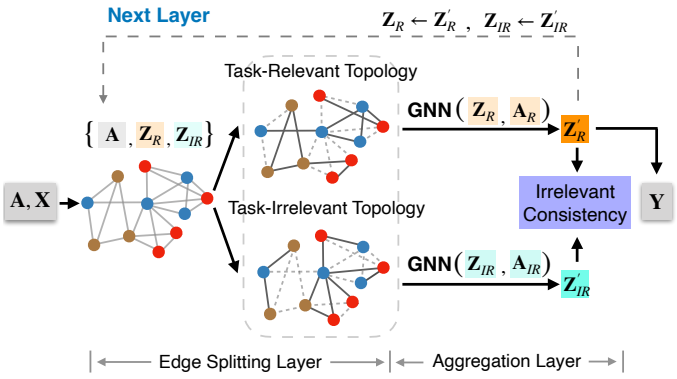


Fig. 2. Illustration of ES-GNN framework where $\mathbf{A}$ and $\mathbf{X}$ denote the adjacency matrix and feature matrix of nodes, respectively. First, $\mathbf{X}$ is projected onto different latent subspaces via different channels R and IR. An edge splitting is then performed to divide the original graph edges into two complementary sets. After that, the node information can be aggregated individually and separately on different edge sets to produce disentangled representations, which are further utilized to make an more accurate edge splitting in the next layer. The task-relevant representation $\mathbf{Z}'_{\mathrm{R}}$ is reasonably granted for prediction, and an Irrelevant Consistency Regularization (ICR) term is developed to further reduce the potential task-harmful information from the final predictive target.

An ES-layer consists of two channels to respectively extract the task-relevant and irrelevant information from nodes. As only the raw feature matrix $\mathbf{X}$ is provided in the beginning, we will project them into two different subspaces before the first ES-layer:

$$\mathbf{Z}_s^{(0)} = \sigma(\mathbf{W}_s^T\mathbf{X} + \mathbf{b}_s), \tag{1}$$

where $\mathbf{W}_s \in \mathbb{R}^{f\times\frac{d}{2}}$ and $\mathbf{b}_s \in \mathbb{R}^{\frac{d}{2}}$ are the learnable parameters in channel $s \in \{\mathrm{R}, \mathrm{IR}\}$, d is the number of node hidden states, and σ is a nonlinear activation function.

Given Hypothesis 1, we adopt a flexible approach for classifying node connections by using continuous edge weights from 0 to 1, reflecting the varying degrees to which edges are task-relevant or irrelevant. Nevertheless, applying metrics to independently determine $\mathbf{A}_{\mathrm{R}}$ and $\mathbf{A}_{\mathrm{IR}}$ based on node similarity may not fully capture the complex interplay between different channels and could diminish the focus on topological distinctions. To address this, for edges where $\mathbf{A}_{(i,j)} = 1$, we parameterize the difference between $\mathbf{A}_{\mathrm{R}(i,j)}$ and $\mathbf{A}_{\mathrm{IR}(i,j)}$, by solving the linear equation:

$$\begin{cases} \mathbf{A}_{\mathrm{R}(i,j)} - \mathbf{A}_{\mathrm{IR}(i,j)} = \alpha_{i,j} \\ \mathbf{A}_{\mathrm{R}(i,j)} + \mathbf{A}_{\mathrm{IR}(i,j)} = 1 \end{cases}.$$

This gives us $\mathbf{A}_{\mathrm{R}(i,j)} = \frac{1+\alpha_{i,j}}{2}$ and $\mathbf{A}_{\mathrm{IR}(i,j)} = \frac{1-\alpha_{i,j}}{2}$ with $-1 \le \alpha_{i,j} \le 1$. To effectively quantify the interaction (or relative importance) between the task-relevant and irrelevant aspects of each edge, we propose a residual scoring mechanism:

$$\alpha_{i,j} = \tanh(\mathbf{g}\left[\mathbf{Z}_{\mathrm{R}[i,:]} \oplus \mathbf{Z}_{\mathrm{IR}[i,:]} \oplus \mathbf{Z}_{\mathrm{R}[j,:]} \oplus \mathbf{Z}_{\mathrm{IR}[j,:]}\right]^T). \tag{2}$$

Here, both of the task-relevant and irrelevant node features are first concatenated and convoluted by learnable

$\mathbf{g} \in \mathbb{R}^{1\times 2d}$, and then passed to the tangent activation function to produce a floating value between -1 and 1. Similar learning scheme can be found in works [18], [26], [27]. To further enhance the distinction between $\mathbf{A}_{\text{R}}$ and $\mathbf{A}_{\text{IR}}$, while acknowledging their inherent continuous nature, one can apply techniques, such as softmax with temperature in Eq. (3), Gumbel-Softmax [90], [91] in Eq. (4), or thresholding in Eq. (5). These methods aim to bring their values closer to 0 or 1, thereby strengthening the clarity of task relevance and promoting graph disentanglement.

$$\mathbf{A}'_{s(i,j)} = \frac{\exp(\mathbf{A}_{s(i,j)}/\tau)}{\sum_{\kappa\in\{\text{R,IR}\}}\exp(\mathbf{A}_{\kappa(i,j)}/\tau)} \tag{3}$$

$$\mathbf{A}'_{s(i,j)} = \frac{\exp((\log(\mathbf{A}_{s(i,j)})+\gamma)/\tau)}{\sum_{\kappa\in\{\text{R,IR}\}}\exp((\log(\mathbf{A}_{\kappa(i,j)})+\gamma)/\tau)} \tag{4}$$

$$\mathbf{A}'_{s(i,j)} = \begin{cases} 1 & \mathbf{A}_{s(i,j)} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $s \in \{\text{R, IR}\}$, τ is a hyper-parameter mediating discreteness degree, and $\gamma \sim \mathit{Gumbel}(0,1)$ is a Gumbel random variable. However, in this work, we find good results without adding any additional discretization techniques, and will leave this investigation to the future work.

4.2 Aggregation Layer

As the split network topologies disclose the partial relations among nodes in different latent spaces, they can be utilized to aggregate information for learning different node aspects. Specifically, we leverage a simple low-pass filter with scaling parameters $\{\epsilon_{\text{R}}, \epsilon_{\text{IR}}\}$ for both task-relevant and irrelevant channels, from the k-th to $k+1$-th layer:

$$\mathbf{Z}_s^{(k+1)} = \epsilon_s \mathbf{Z}_s^{(0)} + (1-\epsilon_s)\mathbf{D}_s^{-\frac{1}{2}}\mathbf{A}_s\mathbf{D}_s^{-\frac{1}{2}}\mathbf{Z}_s^{(k)}. \tag{6}$$

$s \in \{\text{R, IR}\}$ denotes the task-relevant or irrelevant channel, and $\mathbf{D}_s$ is the degree matrix associated with the adjacency matrix $\mathbf{A}_s$. Derivation of Eq. (6) is detailed in our theoretical analysis. Importantly, by incorporating proximity information in different structural spaces, the task-relevant and irrelevant information can be better disentangled in $\mathbf{Z}_{\text{R}}^{(k+1)}$ and $\mathbf{Z}_{\text{IR}}^{(k+1)}$, based on which the next ES-layer can make a more precise partition on the raw topology.

4.3 Irrelevant Consistency Regularization

Stacking ES-layer and aggregation layer iteratively lends itself to disentangling different features of nodes into two distinct representations, denoted by $\mathbf{Z}_{\text{R}}$ and $\mathbf{Z}_{\text{IR}}$. First, $\mathbf{Z}_{\text{R}}$, informed and shaped by $\mathbf{A}_{\text{R}}$, is tuned for prediction, with its development guided by the minimization of the classification loss $\mathcal{L}_{\text{pred}}$. This process not only makes $\mathbf{Z}_{\text{R}}$ predictive of node labels but also implicitly reinforces the task-relevant nature of $\mathbf{A}_{\text{R}}$ via message passing. However, only supervising one channel (R) risks neglecting the meaningfulness of the other (IR), potentially leading to the preservation of erroneous information in predictions. To this end, we introduce a Irrelevant Consistency Regularization (ICR) loss $\mathcal{L}_{\text{ICR}}$, designed to regulate $\mathbf{Z}_{\text{IR}}$ as the opposite of $\mathbf{Z}_{\text{R}}$, i.e., identifying the classification-harmful information within the observed graph. The key rationale is to explore the similarities among nodes that are detrimental to classification tasks within $\mathbf{Z}_{\text{IR}}$. Given any node pairs $(v_i, v_j) \in \mathcal{E}$, we would expect $\mathbf{Z}_{\text{IR}[i,:]}$ and $\mathbf{Z}_{\text{IR}[j,:]}$ to be close in the latent space if they possess different labels, which can be formulated as:

$$\mathcal{L}_{\text{ICR}} = \sum_{(v_i,v_j)\in\mathcal{E}} (1-\delta(\mathbf{y}_i,\mathbf{y}_j))\|\mathbf{Z}_{\text{IR}[i,:]} - \mathbf{Z}_{\text{IR}[j,:]}\|_2^2,$$

where δ is a Kronecker function (1 if $\mathbf{y}_i = \mathbf{y}_j$, 0 otherwise) and $\|\cdot\|_2$ denotes L_2 norm. As such, $\mathbf{Z}_{\text{IR}}$ is constrained with a local consistency between adjacent nodes from different classes, aiding in the exclusion and disentanglement of classification-harmful information from $\mathbf{Z}_{\text{R}}$ and to $\mathbf{Z}_{\text{IR}}$.

Several powerful techniques have been developed to assess label agreement between nodes [27], [92]. In this work, however, we find that using the joint probability from model predictions is effective and eliminates the need for additional trainable parameters. Besides, while the idea in ICR could be adapted to task-relevant representations by making closer nodes with the same predicted label, we avoid this due to the risk of inaccurate prediction during training. Such inaccuracy could irreversibly distort the classification metric space. Instead, we supervise task-irrelevant representations – unused for direct prediction – with noisy

Algorithm 1 Framework of ES-GNN

Input: nodes set: $\mathcal{V}$, edge set: $\mathcal{E}$, adjacency matrix: $\mathbf{A} \in \mathbb{R}^{N\times N}$, node feature matrix: $\mathbf{X} \in \mathbb{R}^{|V|\times F}$, the number of layers: K, scaling parameters: $\{\epsilon_{\text{R}}, \epsilon_{\text{IR}}\}$, irrelevant consistency coefficient: λ_{ICR}, and ground truth labels on the training set: $\{\mathbf{y}_i \in \mathbb{R}^C | \forall v_i \in \mathcal{V}_{\text{trn}}\}$.
Param: $\mathbf{W}_{\text{R}}, \mathbf{W}_{\text{IR}} \in \mathbb{R}^{f\times d}$, $\mathbf{W}_F \in \mathbb{R}^{d\times C}$, $\mathbf{b}_F \in \mathbb{R}^C$, $\{\mathbf{g}^{(k)} \in \mathbb{R}^{1\times 2d} | k = 0, 1, ..., K-1\}$

// Project node features into two subspaces.
for $s \in \{\text{R, IR}\}$ **do**
  $\mathbf{Z}_s^{(0)} \leftarrow \sigma(\mathbf{W}_s^T\mathbf{X} + \mathbf{b}_s)$.
  $\mathbf{Z}_s^{(0)} \leftarrow \text{Dropout}(\mathbf{Z}_s^{(0)})$ *// Enabled only for training.*
end for
// Stack Edge Splitting and Aggregation Layers.
for layer number $k = 0, 1, ..., K-1$ **do**
  // Edge Splitting Layer.
  Initialize $\mathbf{A}_{\text{R}}, \mathbf{A}_{\text{IR}} \in \mathbb{R}^{N\times N}$ with zeros.
  for $(v_i, v_j) \in \mathcal{E}$ **do**
    $\alpha_{i,j} \leftarrow \tanh(\mathbf{g}^{(k)}\left[\mathbf{Z}_{\text{R}[i,:]}^{(k)} \oplus \mathbf{Z}_{\text{IR}[i,:]}^{(k)} \oplus \mathbf{Z}_{\text{R}[j,:]}^{(k)} \oplus \mathbf{Z}_{\text{IR},[j,:]}^{(k)}\right]^T)$.
    $\alpha_{i,j} \leftarrow \text{Dropout}(\alpha_{i,j})$ *// Enabled only for training.*
    $\mathbf{A}_{\text{R}(i,j)} \leftarrow \frac{1+\alpha_{i,j}}{2}$, $\mathbf{A}_{\text{IR}(i,j)} \leftarrow \frac{1-\alpha_{i,j}}{2}$.
  end for
  // Aggregation Layer.
  for $s \in \{\text{R, IR}\}$ **do**
    $\mathbf{Z}_s^{(k+1)} \leftarrow \epsilon_s\mathbf{Z}_s^{(0)} + (1-\epsilon_s)\mathbf{D}_s^{-\frac{1}{2}}\mathbf{A}_s\mathbf{D}_s^{-\frac{1}{2}}\mathbf{Z}_s^{(k)}$.
  end for
end for
// Prediction.
$\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{W}_F^T\mathbf{Z}_{\text{R}[i,:]}^{(K)} + \mathbf{b}_F), \forall v_i \in \mathcal{V}$.
// Optimization with Irrelevant Consistency Regularization.
$\mathcal{L}_{\text{ICR}} = \sum_{(v_i,v_j)\in\mathcal{E}}(1-\delta(\hat{\mathbf{y}}_i,\hat{\mathbf{y}}_j))\|\mathbf{Z}_{\text{IR}[i,:]} - \mathbf{Z}_{\text{IR}[j,:]}\|_2^2$.
$\mathcal{L}_{\text{pred}} = -\frac{1}{|\mathcal{V}_{\text{trn}}|}\sum_{i\in\mathcal{V}_{\text{trn}}}\mathbf{y}_i^T\log(\hat{\mathbf{y}}_i)$.
Minimize $\mathcal{L}_{\text{pred}} + \lambda_{\text{ICR}}\mathcal{L}_{\text{ICR}}$.

labels, offering a margin of error that our proposed layers can correct before reaching the final prediction stage.

4.4 Overall Algorithm

The overall pipeline of ES-GNN is detailed in Algorithm 1. Specifically, we adopt ReLU activation function in Eq. (1) to first map node features into two different channels, and then pass them with the adjacency matrix to an ES-layer for splitting the raw network topology into two complementary parts. After that, these two partial network topologies are utilized to aggregate information in different structural spaces. Alternatively stacking ES-layer and aggregation layer not only enables more accurate disentanglement but also explores the graph information beyond local neighborhood. Finally, a fully connected layer is appended to project the learned representations into class space $\mathbb{R}^C$. We integrate $\mathcal{L}_{\text{ICR}}$ into the optimization process with a irrelevant consistency coefficient λ_{ICR} to have final objective function below, where $\mathcal{L}_{\text{pred}} = -\frac{1}{|\mathcal{V}_{\text{trn}}|}\sum_{v_i\in\mathcal{V}_{\text{trn}}}\mathbf{y}_i^T\log(\hat{\mathbf{y}}_i)$.

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \lambda_{\text{ICR}}\mathcal{L}_{\text{ICR}}. \tag{7}$$

It is noted that the method ES-GNN employs in Eq. (2) for learning edge weights diverges from the L_2 space metrics used in our ICR loss. While parameterizing edges with node similarity in L_2 space seems straightforward, this method is only feasible for modeling task-relevant and irrelevant channels independently. Such an approach may not fully capture the intricate interactions across different channels, possibly reducing the focus on topological distinctions. Alternatively, our strategy employs a flexible attention mechanism, unlike direct metric computation, allowing for the nuanced learning of weight residuals between different channels. Importantly, the use of learnable $\mathbf{g}$ in Eq. (2) lends our method a universal fitting capability, enabling it to adapt and bridge potential inconsistencies between different framework components. This flexibility ensures that our model effectively integrates and responds to the diverse dynamics within the graph structure, maintaining our focus on graph disentanglement.

Finally, we also report in Table 1 the complexity of our ES-GNN in comparison with the baseline models evaluated in the experimental section. Clearly, our model displays the same complexity to FAGCN [18] while being slightly overhead compared to GPR-GNN [11]. Here, we omit the related works, such as GEN [13], WRGAT [14], GloGNN++ [23], ACM-GCN [46], and GOAL [47] as their complexity is obviously higher than others by involving graph reconstruction or node-wise operations.

TABLE 1
Time complexity of the comparison models with one hidden layer as an example. N_e denotes the number of graph aspects assumed in FactorGCN [33], D_{max} represents the maximum node degree, and $|\mathcal{E}_2|$ is the total number of neighbors in the second hop of nodes. Other symbols are earlier defined in the texts.

Models	Complexity
GCN [30]	$\mathcal{O}((f+C)\lvert\mathcal{E}\rvert d)$
GAT [26]	$\mathcal{O}(((2+f)N+(4+C)\lvert\mathcal{E}\rvert)d)$
FactorGCN [33]	$\mathcal{O}(N_eN+(Nf+(3+C)\lvert\mathcal{E}\rvert)d)$
H2GCN [10]	$\mathcal{O}(fd+\lvert\mathcal{E}\rvert D_{\text{max}}+(\lvert\mathcal{E}\rvert+\lvert\mathcal{E}_2\rvert)d)$
FAGCN [18]	$\mathcal{O}(((1+C+f)N+\lvert\mathcal{E}\rvert)d)$
GPR-GNN [11]	$\mathcal{O}((fN+\lvert\mathcal{E}\rvert C)d)$
ES-GNN (Ours)	$\mathcal{O}(((1+C+f)N+\lvert\mathcal{E}\rvert)d)$

5 Theoretical Analysis

In this section, we investigate two important problems: (1) what limits the generalization power of the conventional GNNs on graphs beyond homophily, and (2) how the proposed ES-GNN breaks this limit and performs well on different types of networks. We will answer these questions by first analyzing the typical GNNs as graph signal denoising from a more generalized viewpoint, and then impose our Hypothesis 1 to derive ES-GNN.

5.1 Limited Generalization of Conventional GNNs

Recent studies [8], [9] have proved that most GNNs can be regarded as solving a graph signal denoising problem:

$$\arg\min_{\mathbf{Z}} \ \|\mathbf{Z}-\mathbf{X}\|_2^2 + \xi\cdot tr(\mathbf{Z}^T\mathbf{L}\mathbf{Z}), \tag{8}$$

where $\mathbf{X}\in\mathbb{R}^{N\times F}$ is the input signal, $\mathbf{L}=\mathbf{D}-\mathbf{A}\in\mathbb{R}^{N\times N}$ is the graph Laplacian matrix, and ξ is a constant coefficient. The first term guides $\mathbf{Z}$ to be close to $\mathbf{X}$, while the second term $tr(\mathbf{Z}^T\mathbf{L}\mathbf{Z})$ is the Laplacian regularization, enforcing smoothness between connected nodes. One fundamental assumption made here is that similar nodes should have a higher tendency to connect each other, and we refer it as *standard smoothness assumption* on graphs. However, real-world networks typically exhibit diverse linking patterns of both assortativity and disassortativity. Constraining smoothness on each node pair is prone to mistakenly preserve both of the task-relevant and irrelevant (or even harmful) information for prediction. Given that, we divide the original graph into two subgraphs with the same nodes sets but complementary edge sets, and reformulate Eq. (8) as:

$$\arg\min_{\mathbf{Z}} \ \|\mathbf{Z}-\mathbf{X}\|_2^2 + \xi\cdot tr(\mathbf{Z}^T\mathbf{L}_{\text{R}}\mathbf{Z}) + \xi\cdot tr(\mathbf{Z}^T\mathbf{L}_{\text{IR}}\mathbf{Z}).$$

Here, $\mathbf{L}_{\text{R}}=\mathbf{D}_{\text{R}}-\mathbf{A}_{\text{R}}$, and $\mathbf{L}_{\text{IR}}=\mathbf{D}_{\text{IR}}-\mathbf{A}_{\text{IR}}$, where the task-relevant and irrelevant node relations are separately captured in $\mathbf{A}_{\text{R}}$ and $\mathbf{A}_{\text{IR}}$. Clearly, emphasizing the commonality between adjacent nodes in $\mathbf{A}_{\text{R}}$ is beneficial for keeping task-correlated information only. However, smoothing node pairs in $\mathbf{A}_{\text{IR}}$ simultaneously may preserve classification-harmful similarity between nodes, thus limiting the prediction performance of GNNs.

5.2 Disentangled Smoothness Assumption in ES-GNN

Our Hypothesis 1 suggests that the original graph topology can be partitioned into two complementary ones, wherein connected nodes displays high similarity with either task-relevant or irrelevant features only. We further interpret this result as *disentangled smoothness assumption*, based on which

the conventional graph signal denoising problem in Eq. (8) can be generalized as:

$$\begin{aligned} \arg\min_{\mathbf{Z}_{\text{R}},\mathbf{Z}_{\text{IR}}} \quad & \|\mathbf{Z}_{\text{R}}-\mathbf{X}_{\text{R}}\|_2^2+\|\mathbf{Z}_{\text{IR}}-\mathbf{X}_{\text{IR}}\|_2^2 \\ & +\xi_{\text{R}}\cdot tr(\mathbf{Z}_{\text{R}}^T\mathbf{L}_{\text{R}}\mathbf{Z}_{\text{R}})+\xi_{\text{IR}}\cdot tr(\mathbf{Z}_{\text{IR}}^T\mathbf{L}_{\text{IR}}\mathbf{Z}_{\text{IR}}) \\ \text{where} \quad & \mathbf{L}_{\text{R}}=\mathbf{D}_{\text{R}}-\mathbf{A}_{\text{R}},\mathbf{L}_{\text{IR}}=\mathbf{D}_{\text{IR}}-\mathbf{A}_{\text{IR}} \\ \text{s.t.} \quad & \mathbf{A}_{\text{R}}+\mathbf{A}_{\text{IR}}=\mathbf{A} \\ & \mathbf{A}_{\text{R}(i,j)},\mathbf{A}_{\text{IR}(i,j)}\in[0,1]. \end{aligned} \tag{9}$$

Here, $\mathbf{A}_{\text{R}(i,j)}$ and $\mathbf{A}_{\text{IR}(i,j)}$ measure the degree to which the node connection (v_i,v_j) are relevant and irrelevant to the learning task, respectively. We further name this optimization as *disentangled graph denoising problem*, and finally derive the following theorem:

Theorem 1. *The proposed ES-GNN is equivalent to the solution of the disentangled graph denoising problem in Eq.* (9)*.*

Proof. Let $\mathbf{X}_{\text{R}}\in\mathbb{R}^{\frac{d}{2}}$ and $\mathbf{X}_{\text{IR}}\in\mathbb{R}^{\frac{d}{2}}$ be the results of mapping $\mathbf{X}$ into different channels in Eq. (1), i.e., $\mathbf{X}_{\text{R}}=\mathbf{Z}_{\text{R}}^{(0)}$ and $\mathbf{X}_{\text{IR}}=\mathbf{Z}_{\text{IR}}^{(0)}$. Hypothesis 1 motivates us to define $\mathbf{A}_{\text{R}(i,j)}$ and $\mathbf{A}_{\text{IR}(i,j)}$ as node similarity in two aspects. Combining above constraints, we have a linear system in case of $\mathbf{A}_{(i,j)}=1$:

$$\begin{cases} \mathbf{A}_{\text{R}(i,j)}+\mathbf{A}_{\text{IR}(i,j)}=1 \\ \mathbf{A}_{\text{R}(i,j)}-\mathbf{A}_{\text{IR}(i,j)}=\phi_{\text{res}}(\mathbf{Z}_{\text{R}[i,:]},\mathbf{Z}_{\text{IR}[i,:]},\mathbf{Z}_{\text{R}[j,:]},\mathbf{Z}_{\text{IR}[j,:]}) \end{cases},$$

where $\phi_{\text{res}}(\cdot)$ outputs the residual between $\mathbf{A}_{\text{R}(i,j)}$ and $\mathbf{A}_{\text{IR}(i,j)}$ considering both task-relevant and irrelevant node information, and can be formulated with our residual scoring mechanism in Eq. (2). Solving above equations, we can express both $\mathbf{A}_{\text{R}}$ and $\mathbf{A}_{\text{IR}}$ in terms of $\mathbf{Z}_{\text{R}}$ and $\mathbf{Z}_{\text{IR}}$, i.e.,

$$\mathbf{A}_{\text{R}(i,j)}=\frac{1}{2}(1+\alpha_{i,j}),\ \mathbf{A}_{\text{IR}(i,j)}=\frac{1}{2}(1-\alpha_{i,j}). \tag{10}$$

where $\alpha_{i,j}=\phi_{\text{res}}(\mathbf{Z}_{\text{R}[i,:]},\mathbf{Z}_{\text{IR}[i,:]},\mathbf{Z}_{\text{R}[j,:]},\mathbf{Z}_{\text{IR}[j,:]})$. So far, the optimization problem in Eq. (9) is only made up of variables $\mathbf{X}_{\text{R}}$, $\mathbf{X}_{\text{IR}}$, $\mathbf{Z}_{\text{R}}$, and $\mathbf{Z}_{\text{IR}}$. Directly solving it is still however not easy, as the mixing variables of $\mathbf{Z}_{\text{R}}$ and $\mathbf{Z}_{\text{IR}}$, and the introduced non-linear operator in $\phi_{\text{res}(\cdot)}$ result in a complicated differentiation process.

Instead, we can approach this problem by decoupling the learning of $\mathbf{A}_{\text{R}},\mathbf{A}_{\text{IR}}$ from the optimization target, and employ an alternative learning between stages. Suppose we have attained the task-relevant and irrelevant node features in the k^{th} round, i.e., $\mathbf{Z}_{\text{R}}^{(k)}$ and $\mathbf{Z}_{\text{IR}}^{(k)}$. In the first stage, we can compute $\mathbf{A}_{\text{R}(i,j)}^{(k+1)}$ and $\mathbf{A}_{\text{IR}(i,j)}^{(k+1)}$ using $\{\mathbf{Z}_{\text{R}[i,:]}^{(k)},\mathbf{Z}_{\text{IR}[i,:]}^{(k)},\mathbf{Z}_{\text{R},[j,:]}^{(k)},\mathbf{Z}_{\text{IR}[j,:]}^{(k)}\}$ with Eq. (10), which in fact turns out to be our ES-layer in Section 4.1.

In the second stage, injecting the computed values of $\mathbf{A}_{\text{R}(i,j)}^{(k+1)}$ and $\mathbf{A}_{\text{IR}(i,j)}^{(k+1)}$ relaxes the mixture of variables $\mathbf{Z}_{\text{R}}$ and $\mathbf{Z}_{\text{IR}}$, and the original optimization problem can then be disentangled into two independent targets (as all four penalized terms are positive):

$$\arg\min_{\mathbf{Z}_{\text{R}}^*}\|\mathbf{Z}_{\text{R}}^*-\mathbf{Z}_{\text{R}}^{(0)}\|_2^2+\xi_{\text{R}}\cdot tr(\mathbf{Z}_{\text{R}}^{*T}\mathbf{L}_{\text{R}}^{(k)}\mathbf{Z}_{\text{R}}^*) \tag{11}$$

$$\arg\min_{\mathbf{Z}_{\text{IR}}^*}\|\mathbf{Z}_{\text{IR}}^*-\mathbf{Z}_{\text{IR}}^{(0)}\|_2^2+\xi_{\text{IR}}\cdot tr(\mathbf{Z}_{\text{IR}}^{*T}\mathbf{L}_{\text{IR}}^{(k)}\mathbf{Z}_{\text{IR}}^*) \tag{12}$$

where $\mathbf{L}_{\text{R}}^{(k)}=\mathbf{D}_{\text{R}}^{(k)}-\mathbf{A}_{\text{R}}^{(k)}$ and $\mathbf{L}_{\text{IR}}^{(k)}=\mathbf{D}_{\text{IR}}^{(k)}-\mathbf{A}_{\text{IR}}^{(k)}$ are fixed values. Lemma 1, on the R channel as an example, further shows that our aggregation layer, on the task-relevant and irrelevant topologies, in Section 4.2 is approximately solving these two optimization problems in Eq. (11) and Eq. (12).

Therefore, stacking ES- and aggregation layers iteratively is equivalent to the above alternative learning for solving the *disentangled graph denoising problem* in Eq. (9) with $\mathbf{X}_{\text{R}}=\mathbf{Z}_{\text{R}}^{(0)}$ and $\mathbf{X}_{\text{IR}}=\mathbf{Z}_{\text{IR}}^{(0)}$. Finally, given $\mathbf{Z}_{\text{R}}^{(K)}$ and $\mathbf{Z}_{\text{IR}}^{(K)}$, we minimize the prediction loss $\mathcal{L}_{\text{pred}}$ and the Irrelevant Consistency Regularization $\mathcal{L}_{\text{ICR}}$ in Eq. (7) with Adam [93] algorithm, which imposes concrete meanings on different channels, and simultaneously ensures the convergence of our described alternative learning. □

Lemma 1. *When adopting the normalized Laplacian matrix* $\mathbf{L}_R=\mathbf{I}-\mathbf{D}_R^{-\frac{1}{2}}\mathbf{A}_R\mathbf{D}_R^{-\frac{1}{2}}$*, the feature aggregation operator in Eq.* (6) *with channel* $s=R$ *can be regarded as solving Eq.* (11) *using iterative gradient descent with stepsize* $\beta=\frac{1}{2+2\xi_R}$ *and* $\xi_R=\frac{1}{\epsilon_R}-1$*.*

Proof. We take iterative gradient descent with the stepsize β to solve the denoising problem in Eq. (11) (referred as $\mathcal{L}_{\text{R}}$) as follows:

$$\begin{aligned} \mathbf{Z}_{\text{R}}^{(k+1)} &= \mathbf{Z}_{\text{R}}^{(k)}-\beta\cdot\frac{\partial\mathcal{L}_{\text{R}}}{\partial\mathbf{Z}_{\text{R}}^*}|_{\mathbf{Z}_{\text{R}}^*=\mathbf{Z}_{\text{R}}^{(k)}} \\ &= 2\beta\mathbf{Z}_{\text{R}}^{(0)}+2\beta\xi_{\text{R}}(\mathbf{D}_{\text{R}}^{-\frac{1}{2}}\mathbf{A}_{\text{R}}\mathbf{D}_{\text{R}}^{-\frac{1}{2}})\mathbf{Z}_{\text{R}}^{(k)}+(1-2\beta-2\beta\xi_{\text{R}})\mathbf{Z}_{\text{R}}^{(k)}. \end{aligned}$$

Setting β as $\frac{1}{2+2\xi_{\text{R}}}$ gives us:

$$\mathbf{Z}_{\text{R}}^{(k+1)}=\frac{1}{1+\xi_{\text{R}}}\mathbf{Z}_{\text{R}}^{(0)}+\frac{\xi_{\text{R}}}{1+\xi_{\text{R}}}(\mathbf{D}_{\text{R}}^{-\frac{1}{2}}\mathbf{A}_{\text{R}}\mathbf{D}_{\text{R}}^{-\frac{1}{2}})\mathbf{Z}_{\text{R}}^{(k)},$$

which is equivalent to Eq. (6) while choosing $\xi_{\text{R}}=\frac{1}{\epsilon_{\text{R}}}-1$, i.e.,

$$\mathbf{Z}_{\text{R}}^{(k+1)}=\epsilon_{\text{R}}\mathbf{Z}_{\text{R}}^{(0)}+(1-\epsilon_{\text{R}})(\mathbf{D}_{\text{R}}^{-\frac{1}{2}}\mathbf{A}_{\text{R}}\mathbf{D}_{\text{R}}^{-\frac{1}{2}})\mathbf{Z}_{\text{R}}^{(k)}.$$

□

As the possible classification-harmful similarity between nodes (hidden in $\mathbf{A}_{\text{IR}}$) can be excluded from $\mathbf{Z}_{\text{R}}$ and disentangled in $\mathbf{Z}_{\text{IR}}$ while optimizing Eq. (9), our ES-GNN presents a universal approach that theoretically guarantees good performance on different types of networks.

5.3 Aligning Disentangled and Conventional Problems

It is noted that Eq. (8) can be interpreted as a special case of Eq. (9) in specific graph scenarios. This situation arises in graphs where only edges connecting nodes with identical labels exist, indicating that the similarity between adjacent nodes should be beneficial in predicting their shared label. In such cases, task-irrelevant edges may not exist, as all connections inherently support the task. Consequently, in this scenario, the objective term involving $\mathbf{L}_{\text{IR}}$ in Eq. (9) becomes redundant, amounting to zero, and the need for disentangling $\mathbf{Z}_{\text{R}}$ and $\mathbf{Z}_{\text{IR}}$ is obviated. This leads to the simplification of Eq. (9) into the conventional graph denoising problem Eq. (8), conforming to the standard smoothness assumption across the entire graph.

In practice, completely smooth graphs devoid of edges linking nodes with different labels are rare. Nevertheless, for

TABLE 2
Statistics of real-world datasets.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	F	C	$\mathcal{H}$	$\mathcal{H}_{\text{class}}$	$\mathcal{H}_{\text{adjusted}}$
Squirrel	5,201	217,073	2,089	5	0.22	0.03	0.01
Chameleon	2,227	36,101	2,325	5	0.23	0.06	0.03
Wisconsin	251	499	1,703	5	0.21	0.09	-0.17
Cornell	183	295	1,703	5	0.30	0.05	-0.08
Texas	183	309	1,703	5	0.11	0.00	-0.23
Twtich-DE	9,498	153,138	2,545	2	0.63	0.14	0.14
Actor	7,600	33,544	931	5	0.22	0.01	0.00
Cora	2,708	5,429	1,433	7	0.81	0.77	0.77
Citeseer	3,327	4,732	3,703	6	0.74	0.63	0.67
Pubmed	19,717	44,338	500	3	0.80	0.66	0.69
Polblogs	1,222	16,714	/	2	0.91	0.81	0.81

TABLE 3
Parameters for synthesizing graphs with varying homophily ratios.

$\mathcal{H}_{\text{syn}}$	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
P_{E}	0.02	0.06	0.1	0.2	0.4	0.4	0.6	0.7	0.8	0.9	0.96
P_{I}	0.72	0.81	0.6	0.7	0.9	0.6	0.6	0.45	0.3	0.15	0.045
ω	0.1	0.084	0.1	0.075	0.05	0.062	0.05	0.05	0.05	0.05	0.051

homophilic graphs where most edges connect nodes from the same class, such as in the citation network Cora with homophily ratio 0.81, Eq. (8) serves as a close approximation of Eq. (9). This approximation holds as the term involving $\mathbf{L}_{\text{IR}}$ in Eq. (9) becomes negligible and the inherently classification-harmful information in the graph is almost non-existent. Empirical evidence from Fig. 4 in our study reinforces this understanding, showing that on homophilic graphs like Cora, the majority of informative content is retained in the task-relevant channel, underscoring the minimal presence of classification-harmful information.

6 Experiments

We empirically evaluate our ES-GNN for node classification using both synthetic and real-world datasets in this section.

6.1 Datasets and Experimental Setup

6.1.1 Real-World Datasets

We consider 11 widely used benchmark datasets including both seven heterophilc graphs, i.e., Chameleon, Squirrel [94], Wisconsin, Cornell, Texas [44] (webpage networks), Actor [95] (co-occurrence network), and Twitch-DE [28], [94] (social network), as well as four homophilic graphs including Cora, Citeseer, Pubmed [96] (citation networks), and Polblogs [97], [98] (community network) with statistics shown in Table 2. For Polblogs dataset, since node features are not provided, we use the rows of the adjacency matrix.

6.1.2 Synthetic Data

To investigate the behavior of GNNs on graphs with arbitrary levels of homophily and heterophily, we consider the contextual stochastic block model (CSBM) [99], [100] to construct synthetic graphs with our Hypothesis 1 as guide. The central idea is to define links among nodes under two conditions independently, of which only one is correlated with the classification task. We consider 1,200

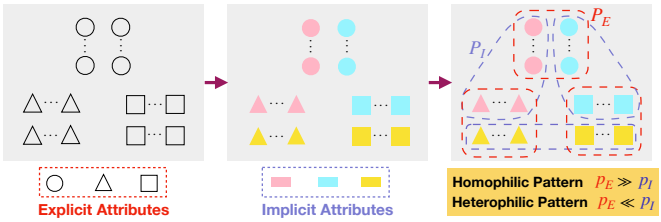


Fig. 3. Synthetic graphs with varying levels of homophily. Node shape and color refer to the explicit and implicit attributes, respectively. Nodes sharing the same shape (or color) are connected with a probability of P_{E} (or P_{I}) and are classified into three categories only based on their different shapes. In this context, "shape" attributes represent task-relevant features, whereas "color" attributes denote task-irrelevant ones. It can be intuitively observed that adequate disentanglement of these attributes is crucial for classification tasks; otherwise, model prediction will inevitably suffer, as misled by the task-irrelevant "color" information.

nodes, 3 equal-size classes, and 500 node features made up of both explicit and implicit attributes. The explicit attributes determine the label assignment, while implicit ones model dependency across different classes. Fig. 3 further illustrates their allocation to nodes with "shape" and "color" as an example. Notably, all these attributes in six types (three explicit and three implicit ones) are randomly sampled from different Gaussian distributions, each pair of them are combined via element-wise addition to attain the final node features. For instance, the features of a node (from class-i) with explicit attribute-i and implicit attribute-j are defined as the addition of two random vectors respectively sampled from $\mathcal{N}(\boldsymbol{\mu}_{\text{E},i}, \boldsymbol{\sigma}_{\text{E},i})$ and $\mathcal{N}(\boldsymbol{\mu}_{\text{I},j}, \boldsymbol{\sigma}_{\text{I},j})$, where $\boldsymbol{\mu}_{\text{E},i}, \boldsymbol{\mu}_{\text{I},j} \in \mathbb{R}^{F_{\text{syn}}}$ are means, $\boldsymbol{\sigma}_{\text{E},i}, \boldsymbol{\sigma}_{\text{I},j} \in \mathbb{R}^{F_{\text{syn}} \times F_{\text{syn}}}$ are the associated covariance matrixes, and $F_{\text{syn}} = 500$ is the feature dimensions. Then, we connect nodes with probability P_{E} if they are from the same class (the task-relevant condition), with probability P_{I} if they share different labels but posses implicit attributes from the same distribution (the task-irrelevant condition). For all other cases, we connect nodes with probability q in a small value, $1\text{e}{-5}$ in this work for ensuring a connected graph. Since no class-imbalance problem exists here, the homophily ratios of our generated graphs are measured using index $\mathcal{H}$. Intuitively, we could anticipate heterophilic connecting pattern when setting $P_{\text{E}} \ll P_{\text{I}}$, and strong homophily otherwise. Quantitatively, the relationship between the homophily ratio $\mathcal{H}_{\text{syn}}$ and parameters P_{E}, P_{I} can be derived with the simple knowledge on combinatorics and statistics while omitting the small value of q: $\mathcal{H}_{\text{syn}}(P_{\text{E}}, P_{\text{I}}) = \frac{3(N_{\text{syn}}-3)}{3(N_{\text{syn}}-3)+2N_{\text{syn}}\frac{P_{\text{I}}}{P_{\text{E}}}}$, with N_{syn} being the total number of nodes. Clearly, we have $\mathcal{H}_{\text{syn}} \to 0$ while $P_{\text{I}} \gg P_{\text{E}}$, and $\mathcal{H}_{\text{syn}} \to 1$ while $P_{\text{I}} \ll P_{\text{E}}$. To avoid possible computational overhead, we also need to control the average node degree of our synthetic graphs. Similarly, we can approximately derive it as the function of P_{I} and P_{E}: $\mathcal{T}(P_{\text{E}}, P_{\text{I}}) = \frac{N_{\text{syn}}-3}{3}P_{\text{E}} + \frac{4N_{\text{syn}}}{9}P_{\text{I}}$. Give this, we have that $\mathcal{H}_{\text{syn}}(\cdot)$ is a function of the fraction between P_{E} and P_{I} with fixed n, and $\mathcal{T}(\cdot)$ is linearly correlated with P_{E} and P_{I}. As such, given fixed P_{E} and P_{I} attaining certain $\mathcal{H}_{\text{syn}}$, we can almost attain the average node degree in any values with a scaling parameter ω, i.e., average degree $= \omega \cdot \mathcal{T}(P_{\text{E}}, P_{\text{I}}) = \mathcal{T}(\omega \cdot P_{\text{E}}, \omega \cdot P_{\text{I}})$ without changing $\mathcal{H}_{\text{syn}}$. In this work, we tune all these parameters such that the

TABLE 4
Node classification accuracies (%) over 100 runs. Error Reduction gives the average improvement of ES-GNN upon baselines w/o Basic GNNs.

Datasets	Heterophilic Graphs							Homophilic Graphs			
	Squirrel	Chameleon	Wisconsin	Cornell	Texas	Twitch-DE	Actor	Cora	Citeseer	Pubmed	Polblogs
GCN [30]	55.2±1.5	67.6±2.0	59.5±3.6	52.8±6.0	61.7±3.7	74.0±1.2	31.2±1.3	79.7±1.2	69.5±1.7	78.7±1.6	89.4±0.9
SGC [6]	50.7±1.3	61.9±2.6	53.7±3.9	51.2±0.9	51.4±2.2	73.9±1.3	30.9±0.6	79.1±1.0	69.9±2.0	76.6±1.3	89.0±1.5
GAT [26]	54.8±2.2	67.3±2.2	57.9±4.5	50.4±5.9	55.4±5.9	73.7±1.3	30.5±1.2	82.0±1.1	69.9±1.7	78.6±2.0	87.4±1.1
NeuralSparse [48]	40.0±1.6	60.5±2.0	70.8±3.4	64.1±5.5	66.4±5.7	71.3±1.3	35.5±1.1	78.5±1.4	69.7±1.8	79.1±1.2	89.3±0.9
GCN-LPA [49]	54.2±1.1	63.4±1.9	63.3±3.7	65.6±7.3	61.2±7.6	74.0±1.2	37.8±0.9	80.4±1.5	69.7±1.7	79.7±1.3	**89.7±0.8**
DisenGCN [66]	42.4±1.6	58.4±2.3	78.1±4.0	77.4±4.4	71.3±5.7	73.5±1.7	36.7±1.2	81.5±1.3	69.2±1.7	80.0±1.6	89.5±0.9
FactorGCN [33]	56.6±2.4	69.8±2.0	64.2±4.8	50.6±1.8	69.5±6.5	73.1±1.4	29.0±1.4	75.2±1.6	61.6±2.0	72.9±2.3	87.9±1.7
VEPM [71]	50.3±1.7	67.3±2.1	55.6±4.9	51.2±7.0	55.8±4.3	73.3±1.2	29.3±1.1	82.2±1.2	69.1±1.9	78.8±1.6	89.5±0.9
DisGNN [72]	55.1±4.8	68.2±1.9	54.6±5.4	52.0±5.7	60.6±3.9	69.2±0.8	30.2±1.3	78.2±1.4	66.2±2.2	77.6±1.7	89.6±0.9
GEN [13]	36.0±4.0	57.6±3.1	83.3±3.6	81.0±3.9	78.3±8.0	74.1±1.4	37.3±1.4	79.8±1.3	69.7±1.6	78.9±1.7	89.6±1.4
WRGAT [14]	39.6±1.4	57.7±1.6	82.9±4.5	79.2±3.5	80.5±6.1	70.0±1.3	38.6±1.1	71.7±1.5	64.1±1.9	73.3±2.1	88.2±1.2
H2GCN [10]	45.1±1.9	62.9±1.9	82.6±4.0	79.6±4.9	79.8±7.3	73.1±1.5	38.4±1.0	81.4±1.4	68.7±2.0	78.0±2.0	89.0±1.0
FAGCN [18]	50.4±2.6	68.9±1.8	82.3±4.4	79.4±5.5	80.3±5.5	74.1±1.4	37.9±1.0	82.6±1.3	70.3±1.6	80.0±1.7	89.3±1.1
GPR-GNN [11]	54.1±1.6	69.6±1.7	82.7±4.1	79.9±5.3	81.7±4.9	74.0±1.6	38.0±1.1	81.5±1.5	69.6±1.7	79.8±1.3	89.5±0.8
GloGNN++ [23]	63.3±1.2	71.4±2.0	84.9±4.2	82.0±3.5	81.4±5.6	72.8±1.1	38.2±1.2	80.9±1.4	70.5±1.9	76.8±2.1	89.6±0.8
ACM-GCN [46]	**67.0±1.3**	**75.3±2.2**	84.3±4.5	82.1±4.9	82.2±5.9	74.2±0.9	36.6±1.0	81.3±1.0	69.4±1.7	79.5±1.4	89.6±0.9
GOAL [47]	57.9±0.9	71.3±2.0	70.5±5.1	54.9±6.6	72.0±7.4	68.5±1.5	36.3±1.0	80.6±1.4	69.7±2.0	78.7±1.3	88.7±1.6
ES-GNN (ours)	62.4±1.4	72.3±2.1	**85.3±4.6**	**82.2±4.0**	**82.3±5.7**	**74.7±1.1**	**38.9±0.8**	**83.0±1.1**	**70.7±1.7**	**80.7±1.4**	**89.7±0.9**
Error Reduction	**11.5%**	**6.4%**	**11.0%**	**11.7%**	**9.4%**	**2.2%**	**3.2%**	**3.3%**	**2.3%**	**2.6%**	**0.5%**

average degree is around 20, and list the experimented values in Table 3.

6.1.3 Data Splitting

For heterophilic graphs and our synthetic graphs, we divides each dataset into 60%/20%/20% corresponding to training/validation/testing to follow [10], [11], [44]. For homophilic graphs, we adopt the popular sparse splitting [6], [26], [30], i.e., 20 nodes per class, 500 nodes, and 1,000 nodes to train, validate, and test models. For each dataset, 10 random splits are created for evaluation.

6.1.4 Baselines

We compare our ES-GNN with 17 baseline models, categorized into four groups: (1) Basic GNNs: GCN [30], SGC [6], and GAT [26]; (2) GNNs prioritizing task-relevance:, NeuralSparse [48], and GCN-LPA [49]; (3) GNNs disentangling graphs: DisenGCN [66], FactorGCN [33], DisGNN [72], and VEPM [71]; (4) GNNs tailored for heterophily: GEN [13], WRGAT [14], H2GCN [10], FAGCN [18], GPR-GNN [11], GloGNN++ [23], ACM-GCN [46], and GOAL [47].

6.1.5 Implementation Details

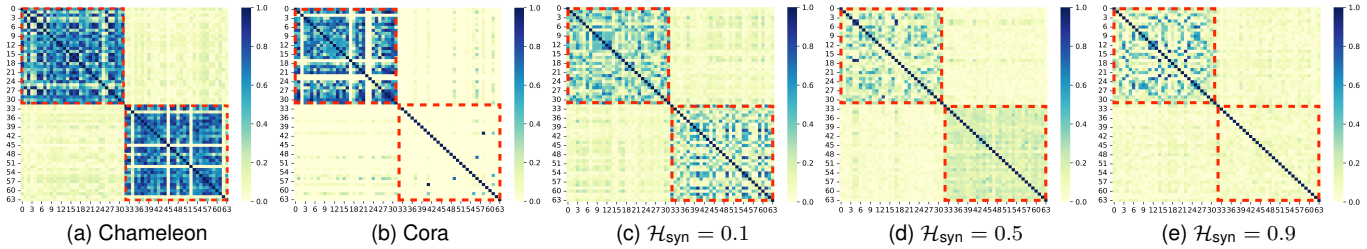
For all the baselines and our model, we set $d = 64$ as the number of hidden states for fair comparison, and tune the hyper-parameters on the validation split of each dataset using Optuna [101] for 200 trials. With the best hyper-parameters, we train models in 1,000 epochs using the early-stopping strategy with a patience of 100 epochs. We then report the models' average performance across 10 runs on the test set for each of the 10 random splits, leading to a total of 100 runs. For reproducibility, we provide the searching space of our hyper-parameters: learning rate $\sim [1\text{e}{-2}, 1\text{e}{-1}]$, weight decay $\sim [1\text{e}{-6}, 1\text{e}{-3}]$, dropout $\sim \{0, 0.1, ..., 0.8\}$ with step 0.1, the number of layers $K \sim \{1, 2, ..., 8\}$ with step 1, scaling parameters $\epsilon_{\text{R}}, \epsilon_{\text{IR}} \sim [5\text{e}{-2}, 0.5]$, and irrelevant consistency coefficient $\lambda_{\text{ICR}} \sim [0, 1]$ for Cora, Citeseer, Pubmed, and Twitch-DE, $[5\text{e}{-8}, 5\text{e}{-6}]$ for Chameleon, Wisconsin, Cornell, and Texas, $[5\text{e}{-5}, 5\text{e}{-3}]$ for Squirrel, and $[5\text{e}{-3}, 5\text{e}{-2}]$ for Actor. Our implementation can be found at https://github.com/jingweio/ES-GNN.

6.2 Results on Real-World Graphs

Table 4 summaries node classification accuracies on real-world datasets over 100 runs with multiple random splits and various model initializations. Generally, our ES-GNN outperforms competitors on most datasets, except for ranking third on Squirrel and second on Chameleon against a wide array of baseline models. In particular, compared to both GNNs specializing in task-relevance, graph disentanglement, and heterophily, our method achieves an average improvements of 11.5%, 6.4%, 11.0%, 11.7%, and 9.4% on heterophilic graphs like Squirrel, Chameleon, Wisconsin, Cornell, and Texas, respectively. On the Twitch-DE and Actor datasets, ES-GNN leads by a smaller margin, with an average increase of 2.7%. In strong homophilic settings, where the majority of edges are intra-class links – essential for node classification – ES-GNN not only capitalizes on these connections but also effectively mitigates the potential noise propagation caused by a small number of inter-class edges. This capability ensures that ES-GNN remains competitive, demonstrating an average performance advantage of 2.2% on the Cora, Citeseer, Pubmed, and Polblogs datasets. In this homophily context, we will further demonstrate the remarkable robustness of ES-GNN in case of perturbation or noisy links in Section 6.6.

6.3 Results on Synthetic Graphs

We examine the learning ability of various models on graphs across the homophily or heterophily spectrum. From Fig. 5, we have the following observations: **1)** Looking through the overall trend, we obtain a "U" pattern on graphs from



(a) Chameleon (b) Cora (c) $\mathcal{H}_{\text{syn}} = 0.1$ (d) $\mathcal{H}_{\text{syn}} = 0.5$ (e) $\mathcal{H}_{\text{syn}} = 0.9$

Fig. 4. Feature correlation analysis. Two distinct patterns (task-relevant and task-irrelevant topologies) can be learned on Chameleon with $\mathcal{H} = 0.23$, while almost all information is retained in the task-relevant channel (0-31) on Cora with $\mathcal{H} = 0.81$. On synthetic graphs in (c), (d), and (e), block-wise pattern in the task-irrelevant channel (32-63) is gradually attenuated with the incremental homophily ratios across 0.1, 0.5, and 0.9. ES-GNN presents one general framework which can be adaptive for both heterophilic and homophilic graphs.

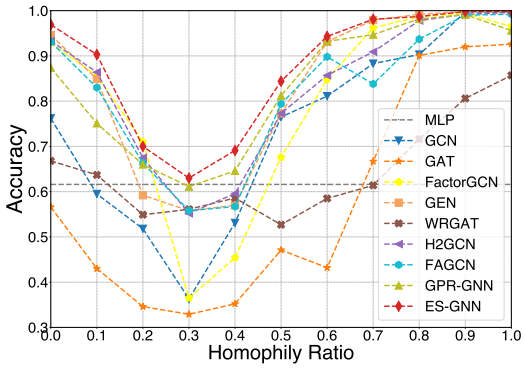


Fig. 5. Results of different models on synthetic graphs with varied homophily ratios, where ES-GNN constantly outperform all the baselines.

TABLE 5
Edge Analysis of our ES-GNN on synthetic graphs with various homophily ratios. "Removed Het." gives the percentage (%) of heterophilic (inter-class) node connections excluded from the task-relevant topology and disentangled in the task-irrelevant topology. The last two rows list the corresponding node classification accuracies (%) of ES-GNN and its variant while ablating ES-layer.

$\mathcal{H}_{\text{syn}}$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**Avg.**
Removed Het.	41.9	53.2	60.8	70.4	74.2	80.7	86.7	87.8	89.9	**71.7**
ES-GNN	90.0	69.6	62.1	69.6	85.4	93.8	98.3	99.2	100.0	**85.3**
ES-GNN w/o ES	84.6	57.9	53.3	53.8	74.2	81.7	86.3	90.4	96.7	**75.4**

the lowest to the highest homophily ratios. That suggests GNNs' prediction performance is not monotonically correlated with graph homophily levels in a strict manner. When it comes to the extreme heterophilic scenario, GNNs tend to alternate node features completely between different classes, thereby still making nodes distinguishable w.r.t. their labels, which coincides with the findings in [102]. **2)** Despite the attention mechanism for adaptively utilizing relevant neighborhood information, GAT turns out to be the least robust method to arbitrary graphs. The entangled information in the mixed assortativity and disassortativity provides weak supervision signals for learning the attention weights. FactorGCN employs a graph factorization to disentangle different graph aspects but still adopts all of them for prediction without judgement, thereby performing poorly especially on the tough cases of $\mathcal{H}_{\text{syn}} = 0.3, 0.4,$ and 0.5. **3)** Both FAGCN and GPR-GNN model the dissimilarity between nearby nodes to go beyond the smoothness assumption in conventional GNNs, and display some superiority under heterophily. However, the correlation between graph edges and classification tasks is not explicitly defined and emphasized in their designs. In other words, the classification-harmful information still could be preserved in their node dissimilarity. Experimental results also show that these methods are constantly beaten by our disentangled approach. **4)** The proposed ES-GNN consistently outperforms, or matches, others across different graphs with different homophily levels, especially in the hardest case with $\mathcal{H}_{\text{syn}} = 0.3$ where some baselines even perform worse than MLP. This is mainly because our ES-GNN is able to distinguish between task-relevant and irrelevant graph links, and makes prediction with the most correlated features only. We further provide detailed analyses in the following sections.

6.4 Correlation Analysis

To better understand our proposed method, we investigate the disentangled features on Chameleon, Cora, and three synthetic graphs as typical examples in Fig. 4. Clearly, on the strong heterophilic graph Chameleon with $\mathcal{H} = 0.23$, correlation analysis of learned latent features displays two clear block-wise patterns, each of which represents task-relevant or task-irrelevant aspect respectively. In contrast, on the citation network Cora with $\mathcal{H} = 0.81$, the node connections are in line with the classification task, since scientific papers mostly cite or are cited by others in the same research topic. Thus, most information will be retained in the task-relevant topology, while very minor information could be disentangled in the task-irrelevant topology (see Fig. 4b). On the other hand, the results on synthetic graphs from Fig. 4c to 4e display an attenuating trend on the second block-wise pattern with the incremental homophily ratios across $0.1, 0.5,$ and 0.9. This correlation analysis empirically verifies that our ES-GNN successfully disentangles the task-relevant and irrelevant features, and also demonstrates its universal adaptivity on different types of networks.

6.5 Edge Analysis

We analyze the split edges from our ES-layer using synthetic graphs as an example in this section. According to Section 6.1.2, the synthetic edges are defined as the task-relevant connections if they link nodes from the same class, and the task-irrelevant ones otherwise. Therefore, we

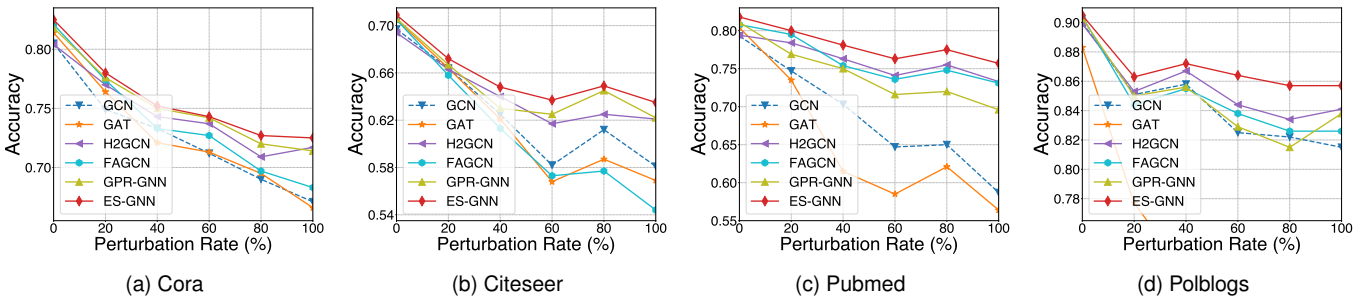


Fig. 6. Results of different models on perturbed homophilic graphs. ES-GNN is able to identify the falsely injected (the task-irrelevant) graph edges, and exclude these connections from the final predictive learning, thereby displaying relative robust performance against adversarial edge attacks.

calculate the percentages of heterophilic node connections, which are excluded from our task-relevant topology and disentangled in the task-irrelevant one, so as to investigate the discerning ability of ES-GNN between edges in different types. As can be observed in Table 5, 71.7% task-irrelevant edges are identified on average across various homophily ratios. On the other hand, we also report the classification accuracies of ES-GNN and its variant while ablating ES-layer, from which approximately 10% degradation can be observed. All of these strongly validate the effectiveness of our ES-layer and reasonably interprets the good performance of ES-GNN.

6.6 Robustness Analysis

By splitting the original graph edge set into task-relevant and task-irrelevant subsets, our proposed ES-GNN enjoys strong robustness particularly on homophilic graphs, since perturbed or noisy aspects of nodes could be purified from the task-relevant topology and disentangled in the task-irrelevant topology. To examine this, we randomly inject fake edges into graphs with perturbed rates from 0% to 100% with a step size of 20%. Adversarially perturbed examples are generated from graphs with strong homophily, such as Cora, Citseer, Pubmed, and Polblogs. As shown in Fig. 6, models considering graphs beyond homophily, i.e., H2GCN, FAGCN, GPR-GNN, and our model, consistently display a more robust behavior than GCN and GAT. That is mainly because fake edges may connect nodes across different labels, and consequently cause erroneous information sharing in the conventional methods.

On the other hand, our ES-GNN beats all the baselines by an average margin of 2% to 3% on Citeseer, Pubmed, and Polblogs while displaying relatively the same results on Cora. We attribute this to the capability of our model in associating node connections with learning tasks. Take Pubmed dataset as an example. We investigate the learned task-relevant topologies and find that 81.0%, 73.0%, 82.1%, 83.0%, 82.6% fake links get removed on adversatial graphs with perturbation rates from 20% to 100%. This also offers evidences supporting that our ES-layer is able to distinguish between task-relevant and irrelevant node connections. Therefore, despite a large number of false edge injections, the proximity information of nodes can still be reasonably mined in our model to predict their labels. Importantly, these empirical results also indicate that ES-GNN can still identify most of the task-irrelevant edges though no clear similarity or association between the connected nodes exists in the adversarial setting.

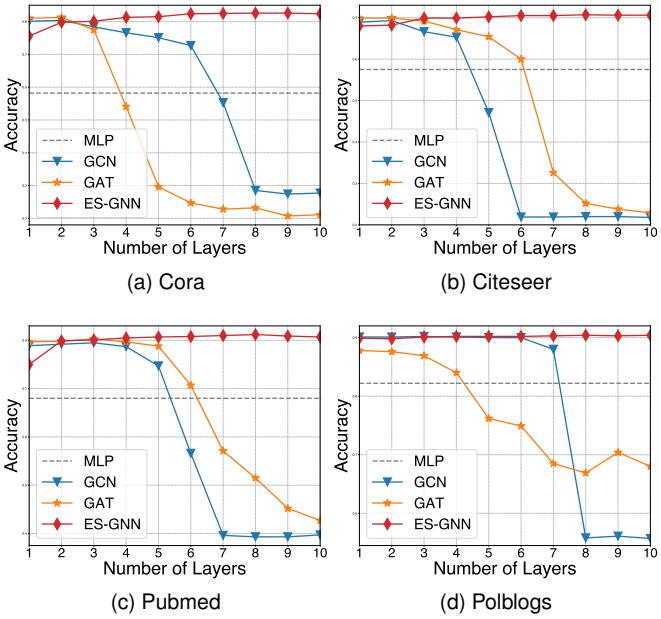


Fig. 7. Classification accuracy vs. model depths.

6.7 Alleviating Over-smoothing Problem

In order to verify whether ES-GNN alleviates the over-smoothing problem, we compare it with GCN and GAT by varying the layer number in Fig. 7. It can be observed that these two baselines attain their highest results when the number of layers reaches around two. As the layer goes deeper, the accuracies of both GCN and GAT gradually drop to a lower point. On the contrary, our ES-GNN presents a stable curve. In spite of starting from a relative lower point, the performance of ES-GNN keeps improving as the model depths increase, and eventually outperforms both GCN and GAT. The main reason is that, our ES-GNN can adaptively utilize proper graph edges in different layers to attain the task-optimal results with enlarged receptive fields. In other words, once an edge stops passing useful information or starts passing harmful messages, ES-GNN tends to identify it and remove it from learning the task-correlated representations, thereby having the ability of mitigating the over-smoothing problem.

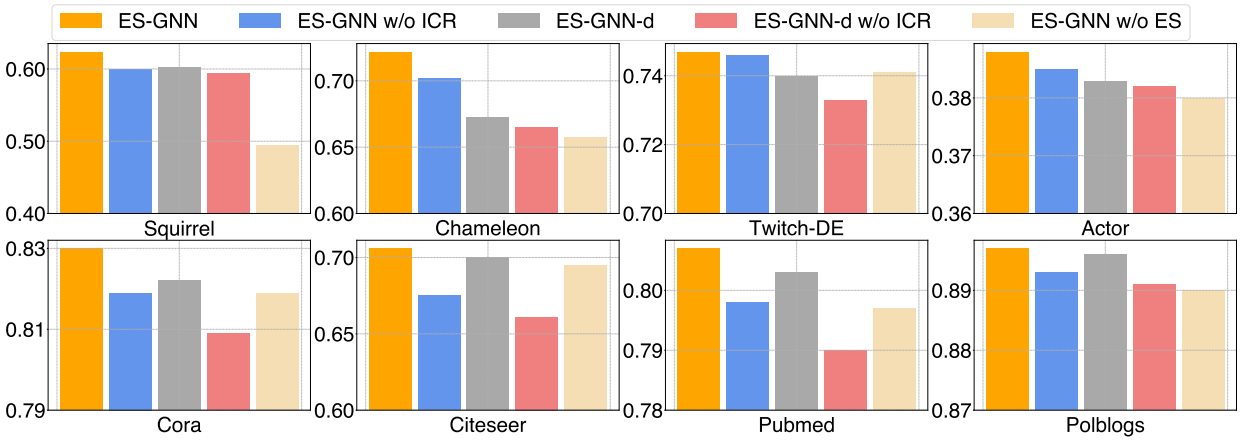


Fig. 8. Ablation study of ES-GNN on eight datasets in node classification.

6.8 Channel Analysis and Ablation Study

In this section, we compare ES-GNN with its variant ES-GNN-d which takes dual (both the task-relevant and irrelevant) channels for prediction, and perform an ablation study. Fig. 8 provides comparison on eight real-world datasets as examples. Here, we first specify some annotations including 1) "w/o ICR": without regularization loss $\mathcal{L}_{\text{ICR}}$, and 2) "w/o ES": without edge splitting (ES-) layer. Overall, two conclusions can be drawn from Fig. 8. First, ES-GNN is consistently better than ES-GNN-d, implying that the task-irrelevant channels indeed capture some false information where model performance downgrades even with the doubled feature dimensions. Second, removing either ICR or ES-layer from both ES-GNN and ES-GNN-d leads to a clear accuracy drop. That validates the effectiveness of our model designs.

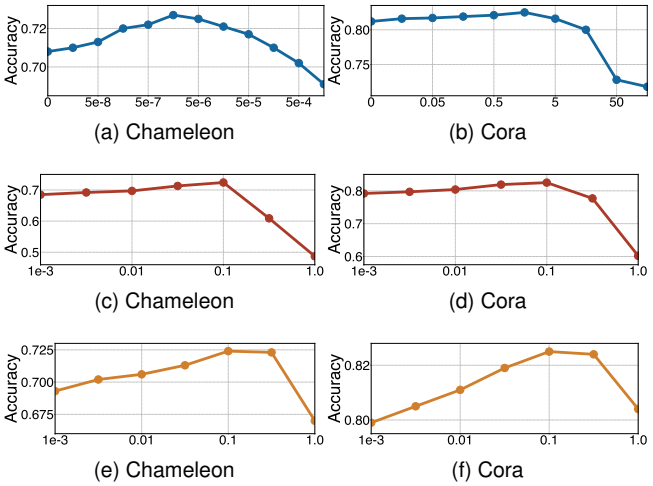


Fig. 9. Sensitivity analysis of hyper-parameters: λ_{ICR}, ϵ_{R} and ϵ_{IR} from top to bottom rows.

6.9 Parameter Study

This section presents the sensitivity analysis of hyper-parameters, specifically λ_{ICR}, ϵ_{R} and ϵ_{IR}, using Chameleon and Cora datasets as typical examples. Fig. 9 illustrates how varying these parameters affect our model's learning performance. Overall, we have the following observations: **1)** The effect of the regularization coefficient λ_{ICR} is depicted in Fig. 9(a)-(b). For instance, the classification accuracy on the Chameleon dataset increases first and then gradually decreases. Favorable results can be attained by choosing λ_{ICR} from $[1\text{e}{-7}, 1\text{e}{-5}]$. A similar trend can be also observed on Cora dataset where λ_{ICR} is relatively robust within a wide albeit distinct interval; **2)** The influence of the scaling coefficients ϵ_{R} and ϵ_{IR} is evaluated by adjusting their values from $1\text{e}{-3}$ to 1.0 on Fig. 9(c)-(d) and Fig. 9(e)-(f), respectively. It can be observed that, despite variations in optimal settings, selecting values between $5\text{e}{-2}$ and 0.5 consistently yields promising performance.

7 Conclusion

In this paper, we develop a novel graph learning framework that enables GNNs to go beyond the strong homophily assumption on graphs. We manage to establish a correlation between node connections and learning tasks through a plausible hypothesis, from which ES-GNN is derived with interpretable edge splitting. Our ES-GNN essentially partitions the original graph structure into task-relevant and irrelevant topologies as guide to disentangle node features, whereby the classification-harmful information can be disentangled and excluded from the final prediction target. Theoretical analysis illustrates our motivation and offers interpretations on the expressive power of ES-GNN on different types of networks. To provide empirical verification, we conduct extensive experiments over 11 benchmark and 1 synthetic datasets. The node classification results demonstrate the overall superior performance of our ES-GNN compared to 15 competitive baselines, which specialize in task-relevance, graph disentanglement and heterophily. In particular, we also conduct analysis on the split edges, correlation among disentangled features, model robustness, and the ablated variants. All of these results demonstrate the success of ES-GNN in identifying graph edges between different types, which also validates the effectiveness of our interpretable edge splitting. In future work, we will further explore more sophisticated designs in the edge splitting layer. Another promising direction would be how to extend

our learning paradigm in accomplishing graph-level tasks.

ACKNOWLEDGMENTS

The work was supported by the following: National Natural Science Foundation of China under No. 92370119, and 62376113.

REFERENCES

[1] G. Ciano, A. Rossi, M. Bianchini, and F. Scarselli, "On inductive–transductive learning with graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 758–769, 2021.

[2] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[3] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1172–1180.

[4] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[5] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1263–1272.

[6] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6861–6871.

[7] M. Balcilar, G. Renton, P. Héroux, B. Gaüzère, S. Adam, and P. Honeine, "Analyzing the expressive power of graph neural networks in a spectral perspective," in *International Conference on Learning Representations*, 2020.

[8] Y. Ma, X. Liu, T. Zhao, Y. Liu, J. Tang, and N. Shah, "A unified view on graph neural networks as graph signal denoising," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1202–1211.

[9] M. Zhu, X. Wang, C. Shi, H. Ji, and P. Cui, "Interpreting and unifying graph neural networks with an optimization framework," in *Proceedings of the Web Conference 2021*, 2021, pp. 1215–1226.

[10] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra, "Beyond homophily in graph neural networks: current limitations and effective designs," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[11] E. Chien, J. Peng, P. Li, and O. Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *International Conference on Learning Representations*, 2021.

[12] J. Zhu, R. A. Rossi, A. Rao, T. Mai, N. Lipka, N. K. Ahmed, and D. Koutra, "Graph neural networks with heterophily," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 11 168–11 176.

[13] R. Wang, S. Mou, X. Wang, W. Xiao, Q. Ju, C. Shi, and X. Xie, "Graph structure estimation neural networks," in *Proceedings of the Web Conference 2021*, 2021, pp. 342–353.

[14] S. Suresh, V. Budde, J. Neville, P. Li, and J. Ma, "Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns," *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

[15] X. Ma, Q. Chen, Y. Ren, G. Song, and L. Wang, "Meta-weight graph neural network: Push the limits beyond global homophily," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1270–1280.

[16] H. Nt and T. Maehara, "Revisiting graph neural networks: All we have is low-pass filters," *arXiv preprint arXiv:1905.09550*, 2019.

[17] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual review of sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[18] D. Bo, X. Wang, C. Shi, and H. Shen, "Beyond low-frequency information in graph convolutional networks," in *AAAI*. AAAI Press, 2021.

[19] M. Liu, Z. Wang, and S. Ji, "Non-local graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[20] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, and D. Koutra, "Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks," *arXiv preprint arXiv:2102.06462*, 2021.

[21] Z. Fang, L. Xu, G. Song, Q. Long, and Y. Zhang, "Polarized graph neural networks," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1404–1413.

[22] L. Yang, W. Zhou, W. Peng, B. Niu, J. Gu, C. Wang, X. Cao, and D. He, "Graph neural networks beyond compromise between attribute and topology," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1127–1135.

[23] X. Li, R. Zhu, Y. Cheng, C. Shan, S. Luo, D. Li, and W. Qian, "Finding global homophily in graph neural networks when meeting heterophily," in *International Conference on Machine Learning*. PMLR, 2022, pp. 13 242–13 256.

[24] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *International conference on machine learning*. PMLR, 2020, pp. 1725–1735.

[25] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *International Conference on Learning Representations*, 2020.

[26] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *International Conference on Learning Representations*, 2018, accepted as poster.

[27] D. Kim and A. Oh, "How to find your friendly neighborhood: Graph attention design with self-supervision," in *International Conference on Learning Representations*, 2021.

[28] D. Lim, F. Hohne, X. Li, S. L. Huang, V. Gupta, O. Bhalerao, and S. N. Lim, "Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 887–20 902, 2021.

[29] O. Platonov, D. Kuznedelev, A. Babenko, and L. Prokhorenkova, "Characterizing graph datasets for node classification: Homophily-heterophily dichotomy and beyond," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[31] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *ICLR*, 2019.

[32] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5453–5462.

[33] Y. Yang, Z. Feng, M. Song, and X. Wang, "Factorizable graph convolutional networks," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[34] K.-H. Lai, D. Zha, K. Zhou, and X. Hu, "Policy-gnn: Aggregation optimization for graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 461–471.

[35] E. Isufi, F. Gama, and A. Ribeiro, "Edgenets: Edge varying graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[36] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[37] Y. Gao, Y. Feng, S. Ji, and R. Ji, "Hgnn+: General hypergraph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3181–3199, 2022.

[38] G. Bouritsas, F. Frasca, S. P. Zafeiriou, and M. Bronstein, "Improving graph neural network expressivity via subgraph isomorphism counting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[39] M. Balcilar, P. Héroux, B. Gauzere, P. Vasseur, S. Adam, and P. Honeine, "Breaking the limits of message passing graph neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 599–608.

[40] L. Faber, A. K. Moghaddam, and R. Wattenhofer, "When comparing to ground truth is wrong: On evaluating gnn explanation methods," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 332–341.

[41] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T.-S. Chua, "Reinforced causal explainer for graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[42] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K. T. Schutt, K.-R. Muller, and G. Montavon, "Higher-order explanations of graph neural networks via relevant walks." *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2021.

[43] Y. Min, F. Wenkel, and G. Wolf, "Scattering gcn: Overcoming oversmoothness in graph convolutional networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 498–14 508, 2020.

[44] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-gcn: Geometric graph convolutional networks," *ArXiv*, vol. abs/2002.05287, 2020.

[45] Y. Hou, J. Zhang, J. Cheng, K. Ma, R. T. Ma, H. Chen, and M.-C. Yang, "Measuring and improving the use of graph information in graph neural networks," in *International Conference on Learning Representations*, 2019.

[46] S. Luan, C. Hua, Q. Lu, J. Zhu, M. Zhao, S. Zhang, X.-W. Chang, and D. Precup, "Revisiting heterophily for graph neural networks," *Advances in neural information processing systems*, vol. 35, pp. 1362–1375, 2022.

[47] Y. Zheng, H. Zhang, V. Lee, Y. Zheng, X. Wang, and S. Pan, "Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs," in *International Conference on Machine Learning*. PMLR, 2023, pp. 42 492–42 505.

[48] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang, "Robust graph representation learning via neural sparsification," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 458–11 468.

[49] H. Wang and J. Leskovec, "Unifying graph convolutional neural networks and label propagation," *arXiv preprint arXiv:2002.06755*, 2020.

[50] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang, "Learning to drop: Robust graph neural network via topological denoising," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 779–787.

[51] S. Zhang, F. Zhu, J. Yan, R. Zhao, and X. Yang, "Dotin: Dropping task-irrelevant nodes for gnns," *arXiv preprint arXiv:2204.13429*, 2022.

[52] P. Trivedi, E. S. Lubana, Y. Yan, Y. Yang, and D. Koutra, "Augmentations in graph contrastive learning: Current methodological flaws & towards better practices," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1538–1549.

[53] X. Gong, C. Yang, and C. Shi, "Ma-gcl: Model augmentation tricks for graph contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4284–4292.

[54] D. Xu, W. Cheng, D. Luo, H. Chen, and X. Zhang, "Infogcl: Information-aware graph contrastive learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 414–30 425, 2021.

[55] Q. Sun, J. Li, H. Peng, J. Wu, X. Fu, C. Ji, and S. Y. Philip, "Graph structure learning with variational information bottleneck," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4165–4174.

[56] M. Yang, Y. Shen, H. Qi, and B. Yin, "Soft-mask: adaptive substructure extractions for graph neural networks," in *Proceedings of the Web Conference 2021*, 2021, pp. 2058–2068.

[57] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15 524–15 543.

[58] C. Wang, H. Zhang, B. Chen, D. Wang, Z. Wang, and M. Zhou, "Deep relational topic modeling via graph poisson gamma belief network," *Advances in Neural Information Processing Systems*, vol. 33, pp. 488–500, 2020.

[59] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[60] I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner, "Towards a definition of disentangled representations," *arXiv preprint arXiv:1812.02230*, 2018.

[61] W. Tang, L. Li, X. Liu, L. Jin, J. Tang, and Z. Li, "Context disentangling and prototype inheriting for robust visual grounding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[62] Y. Dalva, H. Pehlivan, O. I. Hatipoglu, C. Moran, and A. Dundar, "Image-to-image translation with disentangled latent vectors for face editing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[63] S. Chu, D. Kim, and B. Han, "Learning debiased and disentangled representations for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8355–8366, 2021.

[64] C. Chen, M. Ye, M. Qi, and B. Du, "Sketchtrans: Disentangled prototype learning with transformer for sketch-photo recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[65] Y. Liu and X. Liu, "Spoof trace disentanglement for generic face anti-spoofing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3813–3830, 2022.

[66] J. Ma, P. Cui, K. Kuang, X. Wang, and W. Zhu, "Disentangled graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 4212–4221.

[67] Y. Liu, X. Wang, S. Wu, and Z. Xiao, "Independence promoted graph disentangled networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4916–4923.

[68] H. Li, X. Wang, Z. Zhang, Z. Yuan, H. Li, and W. Zhu, "Disentangled contrastive learning on graphs," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 872–21 884, 2021.

[69] T. Xiao, Z. Chen, Z. Guo, Z. Zhuang, and S. Wang, "Decoupled self-supervised learning for graphs," *Advances in Neural Information Processing Systems*, vol. 35, pp. 620–634, 2022.

[70] H. Li, Z. Zhang, X. Wang, and W. Zhu, "Disentangled graph contrastive learning with independence promotion," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[71] Y. He, C. Wang, H. Zhang, B. Chen, and M. Zhou, "A variational edge partition model for supervised graph representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 339–12 351, 2022.

[72] T. Zhao, X. Zhang, and S. Wang, "Exploring edge disentanglement for node classification," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1028–1036.

[73] J. Guo, K. Huang, X. Yi, and R. Zhang, "Learning disentangled graph convolutional networks locally and globally," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[74] S. Fan, X. Wang, Y. Mo, C. Shi, and J. Tang, "Debiasing graph neural networks via learning disentangled causal substructure," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 934–24 946, 2022.

[75] S. Zheng, Z. Zhu, Z. Liu, J. Cheng, and Y. Zhao, "Adversarial graph disentanglement with component-specific aggregation," *IEEE Transactions on Artificial Intelligence*, 2023.

[76] L. Wu, H. Lin, J. Xia, C. Tan, and S. Z. Li, "Multi-level disentanglement graph neural network," *Neural Computing and Applications*, vol. 34, no. 11, pp. 9087–9101, 2022.

[77] X. Guo, L. Zhao, Z. Qin, L. Wu, A. Shehu, and Y. Ye, "Interpretable deep graph generation with node-edge co-disentanglement," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1697–1707.

[78] X. Guo, Y. Du, and L. Zhao, "Deep generative models for spatial networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 505–515.

[79] S. Wang, X. Guo, and L. Zhao, "Deep generative model for periodic graphs," *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[80] G. Mercatali, A. Freitas, and V. Garg, "Symmetry-induced disentanglement on graphs," *Advances in neural information processing systems*, vol. 35, pp. 31 497–31 511, 2022.

[81] Y. Du, X. Guo, H. Cao, Y. Ye, and L. Zhao, "Disentangled spatiotemporal graph generative models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 6, 2022, pp. 6541–6549.

[82] I. Bae and H.-G. Jeon, "Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 911–919.

[83] M. Zhou, "Infinite edge partition models for overlapping community detection and link prediction," in *Artificial intelligence and statistics*. PMLR, 2015, pp. 1135–1143.

[84] L. Xia, Y. Shao, C. Huang, Y. Xu, H. Xu, and J. Pei, "Disentangled graph social recommendation," in *2023 IEEE 39th International Conference on Data Engineering*. IEEE, 2023, pp. 2332–2344.

[85] X. Ren, L. Xia, J. Zhao, D. Yin, and C. Huang, "Disentangled contrastive collaborative filtering," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 1137–1146.

[86] Y. Li, Y. Hao, P. Zhao, G. Liu, Y. Liu, V. S. Sheng, and X. Zhou, "Edge-enhanced global disentangled graph neural network for sequential recommendation," *ACM transactions on knowledge discovery from data*, vol. 17, no. 6, pp. 1–22, 2023.

[87] A. Li, Z. Cheng, F. Liu, Z. Gao, W. Guan, and Y. Peng, "Disentangled graph neural networks for session-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[88] S. Zhao, W. Wei, D. Zou, and X. Mao, "Multi-view intent disentangle graph networks for bundle recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4379–4387.

[89] Y. Qin, X. Wang, Z. Zhang, P. Xie, and W. Zhu, "Graph neural architecture search under distribution shifts," in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 083–18 095.

[90] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[91] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[92] O. Stretcu, K. Viswanathan, D. Movshovitz-Attias, E. A. Platanios, S. Ravi, and A. Tomkins, "Graph agreement models for semi-supervised learning," in *NeurIPS*, 2019.

[93] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[94] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *Journal of Complex Networks*, vol. 9, no. 2, p. cnab014, 2021.

[95] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 807–816.

[96] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Mag.*, vol. 29, pp. 93–106, 2008.

[97] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 us election: Divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*, 2005, pp. 36–43.

[98] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 66–74.

[99] Y. Deshpande, S. Sen, A. Montanari, and E. Mossel, "Contextual stochastic block models," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[100] J. Palowitch, A. Tsitsulin, B. Perozzi, and B. A. Mayer, "Synthetic graph generation to benchmark graph learning," in *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022.

[101] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[102] Y. Ma, X. Liu, N. Shah, and J. Tang, "Is homophily a necessity for graph neural networks?" *ArXiv*, vol. abs/2106.06134, 2021.

Jingwei Guo received the First-class (Hons) degree in Applied Mathematics from University of Liverpool, UK, in 2018. After finishing his undergraduate study, he worked as a Research Associate at Xi'an Jiaotong-Liverpool University of China for a year. He is currently pursing his PhD degree at University of Liverpool, UK. His research focuses on developing new graph neural networks, and applying the techniques in various domains.

Kaizhu Huang (corresponding author) works on machine learning, neural information processing, and pattern recognition. He is currently a tenured Professor of ECE at Duke Kunshan University (DKU). Prof. Huang obtained his PhD degree from Chinese University of Hong Kong (CUHK) in 2004. He worked in Fujitsu Research Centre, CUHK, University of Bristol, National Laboratory of Pattern Recognition, Chinese Academy of Sciences, and Xi'an Jiaotong-Liverpool University from 2004 to 2022. Prof. Huang has been working in machine learning, neural information processing, and pattern recognition. He was the recipient of 2011 Asia Pacific Neural Network Society Young Researcher Award. He received best (runner-up) paper or book awards nine times and published extensively in journals (IEEE T-NNLS, IEEE T-IP, IEEE T-PAMI, IEEE T-CYB) and conferences (AAAI, ICML, CIKM, ECML, ICCV, CVPR, NeurIPS, ICDM). He serves as associated editors/advisory board members in a number of journals and book series. He was invited as keynote speaker at more than 40 international conferences or workshops.

Rui Zhang received the First-class (Hons) degree in Telecommunication Engineering from Jilin University of China in 2001 and the Ph.D. degree in Computer Science and Mathematics from University of Ulster, UK in 2007. After finishing her PhD study, she worked as a Research Associate at University of Bradford and University of Bristol in the UK for 5 years. She joined Xi'an Jiaotong-Liverpool University in 2012 and currently holds the position of Senior Associate Professor. Her research interests include machine learning, data mining and statistical analysis.

Xinping Yi received the Ph.D. degree in electronics and communications from Télécom ParisTech, Paris, France, in 2015. He has been a Professor at Southeast University, Nanjing, China, since 2023. Prior to that, he was a Lecturer (Assistant Professor) with University of Liverpool, Liverpool, UK, a Research Associate with Technische Universitat Berlin, Berlin, Germany, and a Research Assistant with EURECOM, Sophia Antipolis, France. His main research interests include network information theory, trustworthy artificial intelligence, graph machine learning, and their applications in wireless communications.