APPENDIX

Mengzhe Jia, Hanxi Zhu, Zhengpin Li, Ruyu Wang, Jian Wang

School of Data Science, Fudan University

1. THEORY

In this section, we provide a theoretical analysis to validate the effectiveness of GRASP. We focus on its capability to classify nodes with the same label to the correct class. To conduct an evaluation on graph data, we adopt the contextual stochastic block model (cSBM) [1], which is a generative model for graphs with node features and community structure, and widely used in graph learning studies [2, 3].

Definition 1.1 (Contextual Stochastic Block Model (cSBM))**.** *Consider a graph with n nodes divided into C classes (communities). Each node i belongs to a class* c_j, $j \in \{1, 2, \cdots, C\}$. *Edges are generated according to a class-dependent probability matrix* $\mathbf{P} \in [0, 1]^{C\times C}$. *Each node is associated with a feature vector* $\mathbf{x}_i \in \mathbb{R}^F$, *drawn from a class-conditional distribution. Typically, node features are generated as:*

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{c_i}, \mathbf{I}),$$

where $\boldsymbol{\mu}_{c_i} \in \mathbb{R}^F$ *is the mean vector specific to class* c_i.

In this paper, we aim to theoretically analyze how the feature embeddings generated by GRASP—after the graph-splitting and frequency-aware aggregation modules—facilitate correct node classification. To facilitate the theoretical analysis, we adopt the following assumptions. Although they introduce certain simplifications, they are widely used in the literature and do not compromise the general applicability of our results.

Assumption 1. *We limit the classification setting to two classes, i.e.,* $C = 2$.

This simplification is common in theoretical analysis [3], as it allows us to focus on the core mechanisms of the model without getting entangled in the additional complexity introduced by multi-class problem. Importantly, many key insights and derivations in binary settings can be naturally extended to multi-class scenarios through techniques like one-vs-rest. Therefore, this assumption does not limit the relevance of the analysis to real-world applications involving more than two classes.

Corresponding author: jian_wang@fudan.edu.cn

Assumption 2. *The class distribution is balanced, i.e.,* $\mathbb{P}(\mathbf{Y} = 1) = \mathbb{P}(\mathbf{Y} = 0)$.

This assumption helps remove potential confounding factors caused by class imbalance, which can skew statistical estimations and model behavior. By ensuring that both classes are equally represented, we can isolate the effect of structural and frequency-aware design choices on model performance. In practice, even if the data is imbalanced, techniques such as re-weighting or re-sampling can adjust the training process to approximate a balanced setting, further supporting the relevance of this assumption.

Assumption 3. *The aggregated feature embeddings* $\mathbf{h}_i$, $i \in \{1, \cdots, N\}$ *produced by GRASP share the same variance, i.e., each follows a distribution with covariance* $\sigma\mathbf{I}$.

This is a standard assumption in theoretical analysis and is motivated by the fact that GRASP applies the same filtering architecture across all nodes, which tends to normalize the variance of the resulting embeddings. Moreover, assuming the same covariance simplifies the mathematical derivation and allows us to focus on the relative positions of embeddings in the latent space. In real-world settings, normalization layers (e.g., batch/layer norm) and regularization techniques often implicitly enforce similar statistical properties, lending further justification to this assumption.

Then, we establish Theorem 1.1 to give the probability to classify nodes with same label to the same class.

Theorem 1.1. *Under the above assumptions, after propagation in GRASP, for any two nodes u and v from the same class* c_1, *we have:*

$$|\mathbb{P}(\hat{y}_u = c_1|\mathbf{h}_u) - \mathbb{P}(\hat{y}_v = c_1|\mathbf{h}_v)| \leq \frac{1}{\sigma^2} \|\mathbf{h}_u - \mathbf{h}_v\| \cdot \rho. \quad (1)$$

Here, $\mathbb{P}(\hat{y} = c|\mathbf{h})$ *denotes the conditional prediction probability of assigning a node to class c given its feature* $\mathbf{h}$ *given by GRASP .* $\rho = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|$ *denotes the feature separation distance between classes, which reflects the discriminability of mean vectors across communities.*

Proof. GRASP computes feature representations by aggregating information over homophilic and heterophilic subgraphs, formulated respectively as $\mathbf{H}_{\text{hom}} = \frac{1}{2}\left(\mathbf{I} + \hat{\mathbf{A}}_{\text{hom}}\right)\mathbf{X}$ and

$\mathbf{H}_{\text{het}} = \frac{1}{2}\left(\mathbf{I} - \hat{\mathbf{A}}_{\text{het}}\right)\mathbf{X}$. Then following Gaussian distribtiuon, for any node i in class c_1, we obtain:

$$\mathbf{h}_{i,\text{ hom}} \sim \mathcal{N}(\boldsymbol{\mu}_1, \sigma\mathbf{I}),\ \mathbf{h}_{i,\text{ het}} \sim \mathcal{N}\left(\frac{\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2}{2}, \sigma\mathbf{I}\right). \tag{2}$$

Likewise, for a node j from class c_2:

$$\mathbf{h}_{j,\text{ hom}} \sim \mathcal{N}(\boldsymbol{\mu}_2, \sigma\mathbf{I}),\ \mathbf{h}_{j,\text{ het}} \sim \mathcal{N}\left(\frac{\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1}{2}, \sigma\mathbf{I}\right). \tag{3}$$

GRASP then combines the two representations into the final embedding via a weighted sum $\mathbf{H} = \tau\mathbf{H}_{\text{hom}} + (1-\tau)\mathbf{H}_{\text{het}}$:

$$\begin{aligned}
&\mathbf{h}_i \sim \mathcal{N}\left(\frac{1+\tau}{2}\boldsymbol{\mu}_1 - \frac{1-\tau}{2}\boldsymbol{\mu}_2,\ \sigma^{'}\mathbf{I}\right),\ \text{for } i \text{ belongs to class } c_1,\\
&\mathbf{h}_j \sim \mathcal{N}\left(\frac{1+\tau}{2}\boldsymbol{\mu}_2 - \frac{1-\tau}{2}\boldsymbol{\mu}_1,\ \sigma^{'}\mathbf{I}\right),\ \text{for } j \text{ belongs to class } c_2,
\end{aligned} \tag{4}$$

where $\sigma^{'} = \sqrt{\tau^2 + (1-\tau)^2}\sigma$. Here we denote $\boldsymbol{\mu}_1^{'} = \frac{1+\tau}{2}\boldsymbol{\mu}_1 - \frac{1-\tau}{2}\boldsymbol{\mu}_2, \boldsymbol{\mu}_2^{'} = \frac{1+\tau}{2}\boldsymbol{\mu}_2 - \frac{1-\tau}{2}\boldsymbol{\mu}_1$.

The conditional probability that node u from class c_1 is predicted as c_1 by GRASP, given its embedding $\mathbf{f}_u$, can be computed via Bayes' rule:

$$\mathbb{P}(\hat{y}=c_1|\mathbf{h}_u) = \frac{\mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_1)\mathbb{P}(\hat{y}_u=c_1)}{\mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_1)\mathbb{P}(\hat{y}_u=c_1) + \mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_2)\mathbb{P}(\hat{y}_u=c_2)}. \tag{5}$$

Since we assume a balanced class prior with $\mathbb{P}(\mathbf{Y}=1) = \mathbb{P}(\mathbf{Y}=0)$ as Assumption 2 shows,

$$\begin{aligned}
\mathbb{P}(\hat{y}=c_1|\mathbf{h}_u) &= \frac{\mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_1)}{\mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_1) + \mathbb{P}(\mathbf{h}_u|\hat{y}_u=c_2)}\\
&= \frac{\exp\left(-\frac{(\mathbf{h}_u-\boldsymbol{\mu}_1^{'})^2}{2\sigma^{'2}}\right)}{\exp\left(-\frac{(\mathbf{h}_u-\boldsymbol{\mu}_1^{'})^2}{2\sigma^{'2}}\right) + \exp\left(-\frac{(\mathbf{h}_u-\boldsymbol{\mu}_2^{'})^2}{2\sigma^{'2}}\right)}.
\end{aligned} \tag{6}$$

To make the equation more compact and readable, we define

$$\begin{aligned}
\mathbf{z}_{u1} &= \left(\mathbf{h}_u - \boldsymbol{\mu}_1^{'}\right)^2,\ \mathbf{z}_{u2} = \left(\mathbf{h}_u - \boldsymbol{\mu}_2^{'}\right)^2,\\
\mathbf{z}_{v1} &= \left(\mathbf{h}_v - \boldsymbol{\mu}_1^{'}\right)^2,\ \mathbf{z}_{v2} = \left(\mathbf{h}_v - \boldsymbol{\mu}_2^{'}\right)^2,
\end{aligned} \tag{7}$$

and rewrite the probability expression (Eq. 6) accordingly as

$$\mathbb{P}(\hat{y}=c_1|\mathbf{h}_u) = \frac{\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right)}{\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right) + \exp\left(-\frac{\mathbf{z}_{u2}}{2\sigma^{'2}}\right)}. \tag{8}$$

Consequently, the difference in classification probability between nodes u and v from the same class is:

$$\begin{aligned}
&|\mathbb{P}(\hat{y}_u=c_1|\mathbf{h}_u) - \mathbb{P}(\hat{y}_v=c_1|\mathbf{h}_v)|\\
&= \left|\frac{\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right)}{\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right) + \exp\left(-\frac{\mathbf{z}_{u2}}{2\sigma^{'2}}\right)} - \frac{\exp\left(-\frac{\mathbf{z}_{v1}}{2\sigma^{'2}}\right)}{\exp\left(-\frac{\mathbf{z}_{v1}}{2\sigma^{'2}}\right) + \exp\left(-\frac{\mathbf{z}_{v2}}{2\sigma^{'2}}\right)}\right|\\
&= \left|\frac{\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right)\exp\left(-\frac{\mathbf{z}_{v2}}{2\sigma^{'2}}\right) - \exp\left(-\frac{\mathbf{z}_{u2}}{2\sigma^{'2}}\right)\exp\left(-\frac{\mathbf{z}_{v1}}{2\sigma^{'2}}\right)}{\left[\exp\left(-\frac{\mathbf{z}_{u1}}{2\sigma^{'2}}\right) + \exp\left(-\frac{\mathbf{z}_{u2}}{2\sigma^{'2}}\right)\right]\left[\exp\left(-\frac{\mathbf{z}_{v1}}{2\sigma^{'2}}\right) + \exp\left(-\frac{\mathbf{z}_{v2}}{2\sigma^{'2}}\right)\right]}\right|.
\end{aligned} \tag{9}$$

Since each exponential term lies in $[0, 1]$, the denominator is bounded above by 4. Letting it be denoted by $\exp(m)$, we apply the Lagrange mean value theorem and derive:

$$\begin{aligned}
&|\mathbb{P}(\hat{y}_u=c_1|\mathbf{h}_u) - \mathbb{P}_2(\hat{y}_v=c_1|\mathbf{h}_v)|\\
&= \left|\exp\left(-\frac{(\mathbf{z}_{u1}+\mathbf{z}_{v2})}{2\sigma^{'2}} - m\right) - \exp\left(-\frac{(\mathbf{z}_{u2}+\mathbf{z}_{v1})}{2\sigma^{'2}} - m\right)\right|\\
&\le \frac{1}{2\sigma^{'2}}\left|-(\mathbf{z}_{u1}+\mathbf{z}_{v2}) + (\mathbf{z}_{u2}+\mathbf{z}_{v1})\right|\\
&= \frac{1}{2\sigma^{'2}}\left|-\left(\left(\mathbf{h}_u - \boldsymbol{\mu}_1^{'}\right)^2 + \left(\mathbf{h}_v - \boldsymbol{\mu}_2^{'}\right)^2\right) + \left(\left(\mathbf{h}_u - \boldsymbol{\mu}_2^{'}\right)^2 + \left(\mathbf{h}_v - \boldsymbol{\mu}_1^{'}\right)^2\right)\right|\\
&= \frac{1}{\sigma^{'2}}\|\mathbf{h}_u - \mathbf{h}_v\| \cdot \|\boldsymbol{\mu}_1^{'} - \boldsymbol{\mu}_2^{'}\|\\
&= \frac{1}{\sigma^{'2}}\|\mathbf{h}_u - \mathbf{h}_v\| \cdot \|\frac{1+\tau}{2}\boldsymbol{\mu}_1 - \frac{1-\tau}{2}\boldsymbol{\mu}_2 - \frac{1+\tau}{2}\boldsymbol{\mu}_2 + \frac{1-\tau}{2}\boldsymbol{\mu}_1\|\\
&= \frac{1}{\sigma^{'2}}\|\mathbf{h}_u - \mathbf{h}_v\| \cdot \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|.
\end{aligned} \tag{10}$$

This completes the proof. □

Theorem 1.1 shows that the consistency of predictions for nodes within the same class is affected by both feature similarity and label similarity. In contrast, as pointed out in [3], for traditional GNNs such as SGC [4], the probability difference between predicting two nodes as belonging to class c_1 is bounded by

$$|\mathbb{P}(\hat{y}_u=c_1|\mathbf{h}_u) - \mathbb{P}(\hat{y}_v=c_1|\mathbf{h}_v)| \le \frac{\rho}{\sqrt{2\pi}\sigma}\left(\|\mathbf{h}_u - \mathbf{h}_v\| + \rho\,|e_u - e_v|\right), \tag{11}$$

where the SGC model follows a simplified propagation rule $\hat{\mathbf{Y}} = \text{softmax}(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^k)$, removing non-linear activation functions between layers. Here, $e_i = \frac{|j\in\mathcal{N}(i):y_j=y_i|}{d_i}$ denotes the local homophily ratio of node i. Compared with Eq. 11, GRASP achieves a tighter upper bound on the probability difference, indicating a higher likelihood that nodes from the same class will be classified consistently. This improvement stems from GRASP's ability to construct homophilic and heterophilic subgraphs, where nodes exhibit more uniform local homophily. As a result, the model generates more

Table 1: The statistics of the datasets

Dataset	#Nodes	#Train	#Val	#Test	#Edges	#Feat.	#Classes
Cora	2708	1625	541	542	5429	1433	7
Citeseer	3327	1996	665	666	4732	3703	6
Pubmed	19717	11830	3943	3944	44338	500	3
Chameleon	2277	1366	455	456	36101	2325	5
Cornell	183	109	37	37	298	1703	5
Texas	183	109	37	37	309	1703	5
Squirrel	5201	3121	1040	1040	217073	2089	5

discriminative and robust embeddings, ultimately leading to improved classification performance. Overall, this theoretical result provides strong support for the core design of GRASP's graph-splitting strategy, highlighting it as a principled and effective approach to addressing heterophily in graph-structured data.

2. EXPERIMENT

2.1. Datasets, Baselines, and Setups

In this section, we conduct experiments to evaluate the performance of GRASP on node classification across various graph benchmarks. Specifically, we consider two types of datasets, and the dataset statistics details can be found in Tab. 1.

- Homophilic graphs: Three citation networks Cora, Citeseer, and PubMed [5]. In these graphs, nodes represent research papers, and edges denote citation relationships between them. Node features are extracted from the bag-of-words representations of the paper content, while labels correspond to the research fields. These datasets are characterized by strong homophily, meaning that connected nodes often belong to the same class. This structure makes traditional GNNs effective, as they can leverage low-frequency information to aggregate similar features for central nodes.
- ii) Heterophilic graphs: Chameleon, Squirrel [6], Cornell, and Texas [7]. Chameleon and Squirrel are Wikipedia-based networks where nodes represent web pages on specific topics, and edges correspond to hyperlinks between them. Node features are derived from the informative nouns in the text, and labels indicate the traffic levels of these pages. Cornell and Texas are subsets of the WebKB dataset, where nodes represent web pages from computer science departments of different universities, and edges indicate mutual links between them. These graphs exhibit strong heterophily, and further challenge traditional GNNs, as the aggregation mechanism tends to mix information from dissimilar nodes.

We compare our approach with three categories of baseline methods. These baselines serve as strong comparisons to evaluate the effectiveness of GRASP in handling heterophilic graphs.

- Traditional GNNs: We include Graph Convolutional Networks (GCN) [8] and GraphSAGE [9], which are widely adopted homophilic GNNs. They follow the uniform message-passing framework, aggregating information from one-hop neighbors while assuming that connected nodes share similar labels. This works well for homophilic graphs but struggles on heterophilic ones due to the mixture of dissimilar node features.
- Graph Structure Learning (GSL) methods designed under heterophilic scenarios: GNNs are highly sensitive to the quality of the underlying graph, and real-world graphs often contain suboptimal connections, such as heterophilic edges that hinder effective learning. Graph Structure Learning addresses this issue by optimizing the graph structure during training. This category includes WRGAT [10] and GOAL [11], which aim to improve the given graph structure before applying graph neural networks.
- Heterophilic spectral-based GNNs: Unlike traditional GNNs, which primarily rely on spatial message passing, spectral-based methods focus on learning more expressive filters in the graph spectral domain, including FAGCN [6], GPR-GNN [12], BernNet [13], ACM-GNN [14], UniFilter [15], and PC-Conv [16].

To evaluate the performance of our proposed method, we conduct experiments on a node classification task. We randomly split the set of nodes into training, validation, and test sets, following a standard ratio of 60%, 20%, and 20%, respectively. We report the mean and standard deviation of the classification accuracy over ten independent runs to ensure the robustness of our findings. All experiments are conducted on two NVIDIA GeForce RTX 3090 GPUs, each with 24GB of memory. We employ a random search strategy to identify the best hyperparameters for each dataset. Specifically, we define a search space for key hyperparameters and randomly sample configurations during training. This approach allows us to explore different parameter settings efficiently without introducing human bias.

2.2. Node Classification Results

In Tab. 2, we present the results of node classification experiments conducted on seven widely-used benchmark datasets. As can be seen, our proposed GRASP outperforms the three classes of methods on most datasets, highlighting its effectiveness on both homophilic and heterophilic graphs. Several key factors contribute to the superior performance of GRASP. First, unlike graph structure learning methods

Table 2: Node classification results on seven real-world benchmark datasets. We highlight the best results in bold and the second-best results by underlining.

Dataset	Homophilic			Heterophilic				Avg Rank
	Cora	Citeseer	PubMed	Chameleon	Cornell	Texas	Squirrel	
GCN	87.78 ± 0.96	81.39 ± 1.23	88.90 ± 0.32	64.18 ± 2.62	82.46 ± 3.11	83.11 ± 3.20	44.76 ± 1.39	7.4
GraphSAGE	86.58 ± 0.26	78.24 ± 0.30	86.85 ± 0.11	62.15 ± 0.42	71.41 ± 1.24	79.03 ± 1.20	41.26 ± 0.26	9.9
WRGAT	88.20 ± 1.26	76.81 ± 1.89	88.52 ± 0.92	65.24 ± 0.87	81.62 ± 3.90	83.63 ± 5.50	48.85 ± 0.78	8.1
GOAL	88.75 ± 0.87	77.15 ± 0.95	89.25 ± 0.55	71.65 ± 1.66	84.90 ± 0.66	92.02 ± 0.74	60.53 ± 1.60	5.0
FAGCN	88.85 ± 1.36	82.37 ± 1.46	89.98 ± 0.54	65.47 ± 2.84	88.03 ± 5.60	88.85 ± 4.39	42.24 ± 1.20	5.1
GPR-GNN	79.51 ± 0.36	67.63 ± 0.38	85.07 ± 0.09	67.48 ± 0.40	91.36 ± 0.70	92.92 ± 0.61	49.93 ± 0.53	7.6
BernNet	88.52 ± 0.95	80.09 ± 0.79	88.48 ± 0.41	68.29 ± 1.58	92.13 ± 1.64	93.12 ± 0.65	51.35 ± 0.73	4.9
ACM-GNN	87.64 ± 1.22	80.93 ± 0.97	88.79 ± 0.50	60.48 ± 1.55	93.77 ± 2.77	93.44 ± 2.54	40.91 ± 1.31	6.6
UniFilter	88.28 ± 1.36	77.98 ± 5.28	91.24 ± 0.69	73.63 ± 2.03	85.74 ± 3.36	89.02 ± 3.52	**65.75 ± 0.96**	4.4
PC-Conv	88.13 ± 0.43	79.97 ± 0.98	89.19 ± 0.41	65.79 ± 1.47	81.48 ± 2.46	93.37 ± 0.82	51.57 ± 1.08	5.9
GRASP	**90.77 ± 0.30**	**82.80 ± 0.43**	**91.40 ± 0.55**	**73.78 ± 2.27**	**93.82 ± 1.20**	**94.08 ± 2.64**	64.75 ± 1.71	1.1

such as GOAL, GRASP splits the original graph into two parts without adding new edges, thereby reducing the potential noise. Second, compared to spectral-based methods that extract information from the spectrum with all frequency information mixed, GRASP adopts a more flexible mechanism. It explicitly separates low- and high-frequency signals via learned homophilic and heterophilic subgraphs, allowing simple and efficient filters to capture the most relevant information. By aligning the filtering process with the underlying structural properties of the graph, GRASP is able to adaptively emphasize frequency components that are most informative for a given task. This alignment not only enhances the interpretability of the model but also provides a unified framework that works well across graphs with varying degrees of homophily. In contrast, many baseline methods are either over-specialized to homophilic settings or fail to adequately differentiate different frequency signals, particularly in heterophilic contexts. The consistently superior average ranking across datasets further demonstrates that GRASP can produce highly expressive node embeddings, which directly contribute to improved classification outcomes. This underscores the strength of its frequency-aware architecture in effectively handling both homophilic and heterophilic graph structures in real-world applications.

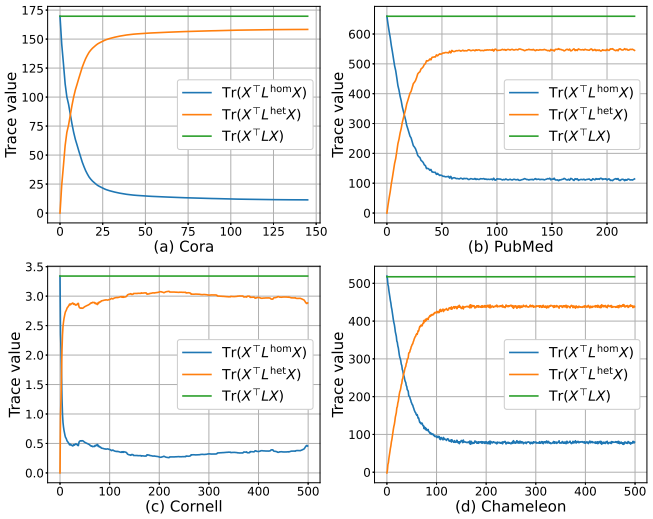

Fig. 1: The behavior of embedding smoothness and roughness regularizers during training on Cora, PubMed, Cornell and Chameleon. The x-axis stands for the training epochs.

2.3. Model Analysis

2.3.1. Trace Curve

Fig. 1 shows the behavior of the embedding smoothness $\mathcal{L}_{\text{ES}}$ and roughness $\mathcal{L}_{\text{ER}}$, during the training process on for example datasets Cora, Pubmed, Cornell and Chameleon. As the training progresses, the embeddings on the homophilic subgraphs become smoother, while those on the heterophilic subgraphs become rougher. This observation highlights GRASP's ability to effectively split the original graph into homophilic and heterophilic subgraphs and thus decouple different frequency information, successfully .

2.3.2. Ablation Study

To better understand the impact of each regularizer on model performance, we conduct the following ablation studies in Tab. 3. w/o rel removes the relational matrices learning and directly learn two $N \times N$ adjacency matrices. $\mathcal{L}_{\text{F}}$-hom and $\mathcal{L}_{\text{F}}$-het stand for only learn the homophilic and the heterophilic subgraph $\mathbf{A}^{\text{hom}}$ or $\mathbf{A}^{\text{het}}$ respectively, while the other subgraph is obtained by subtraction, i.e., $\mathbf{A}^{\text{het}} = \mathbf{A} - \mathbf{A}^{\text{hom}}$ or $\mathbf{A}^{\text{hom}} = \mathbf{A} - \mathbf{A}^{\text{het}}$. w/o $\mathcal{L}_{\text{ES}}$ and $\mathcal{L}_{\text{ER}}$ remove either the embedding smoothness or roughness, allowing subgraphs to be generated purely based on the model's adaptive learning.

Table 3: Ablation study. We highlight the best results in bold.

Ablation	Cora	Pubmed	Cornell	Chameleon
w/o rel	90.01	89.30	93.42	73.67
$\mathcal{L}_{\text{F}}$-hom	89.11	91.36	85.94	69.54
$\mathcal{L}_{\text{F}}$-het	84.31	90.89	90.79	69.92
w/o $\mathcal{L}_{\text{ES}}$	85.06	88.10	83.78	65.29
w/o $\mathcal{L}_{\text{ER}}$	87.82	90.03	84.68	68.82
GRASP	**90.77**	**91.40**	**93.82**	**73.78**

On all datasets, removing each component results in a clear performance drop. These observations affirm the importance of each component, showing that their combination enables GRASP to achieve more accurate node embeddings and superior downstream performance.

2.3.3. Parameters Sensitive Analysis

In this subsection, we conduct a sensitivity analysis on several key hyper-parameters in our model, including weight decay, Frobenius norm weight (α), smoothness and roughness regularizers for the homophilic and heterophilic subgraphs (β and γ), the balance between low- and high-pass filters (τ), hidden dimension size, learning rate of GNN (η), and the learning rate of the adjacency matrix ($\eta^{'}$). To ensure fair evaluation, we fix all other hyper-parameters to their optimal values when tuning a specific parameter, and the experiments are performed on the two homophilic and two heterophilic graphs: Cora, Pubmed, Cornell and Squirrel. Specifically, weight decay is varied from 5e-6 to 5e-4, α, β, and γ from 5e-6 to 5e-2, τ from 0 to 1, hidden size from 16 to 128, and both η and $\eta^{'}$ from 5e-4 to 5e-2.

The experimental results are illustrated in Fig. 2. To better understand the impact of key components in GRASP, we conduct a brief analysis on the core hyperparameters. The Frobenius norm weight α leads to the best result at a moderate value, while overly small or large values harm performance, as they either under-regularize or over-constrain the graph structure learning. Both β and γ show clear trends: increasing their values improves accuracy up to a point, with the best performance observed, validating the importance of smoothness and roughness regularization. For the mixing ratio between low- and high-pass filters τ, the result indicates that balancing homophilic and heterophilic information leads to better representations. Too much emphasis on either type of information results in degraded performance. Overall, these results highlight the significance of careful hyper-parameter tuning and confirm that each component—regularization, smoothness, and frequency balance—plays a vital role in GRASP's capability of learning meaningful embeddings.

3. REFERENCES

[1] Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel, "Contextual stochastic block models," in *Advances in Neural Information Processing Systems*, 2018.

[2] Xinyi Wu, Zhengdao Chen, William Wang, and Ali Jadbabaie, "A non-asymptotic analysis of oversmoothing in graph neural networks," *arXiv preprint arXiv:2212.10701*, 2022.

[3] Haitao Mao, Zhikai Chen, Wei Jin, Haoyu Han, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang, "Demystifying structural disparity in graph neural networks: Can one size fit all?," in *Proceedings of the 37th Conference on Neural Information Processing Systems*, 2023, pp. 37013–37026.

[4] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger, "Simplifying graph convolutional networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, vol. 97, pp. 6861–6871.

[5] Prithviraj Sen, Galen Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[6] Dong Bo, Xiao Wang, Chuan Shi, and Huawei Shen, "Beyond low-frequency information in graph convolutional networks," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021, pp. 3950–3957.

[7] Hongbin Pei, Bi Wei, King C. Chang, Yanfeng Lei, and Bo Yang, "Geom-gcn: Geometric graph convolutional networks," in *International Conference on Learning Representations*, 2020.

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

[9] William L. Hamilton, Rex Ying, and Jure Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.

[10] Saba Suresh, Victoria Budde, Jennifer Neville, Pan Li, and Jing Ma, "Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021, pp. 1541–1551.

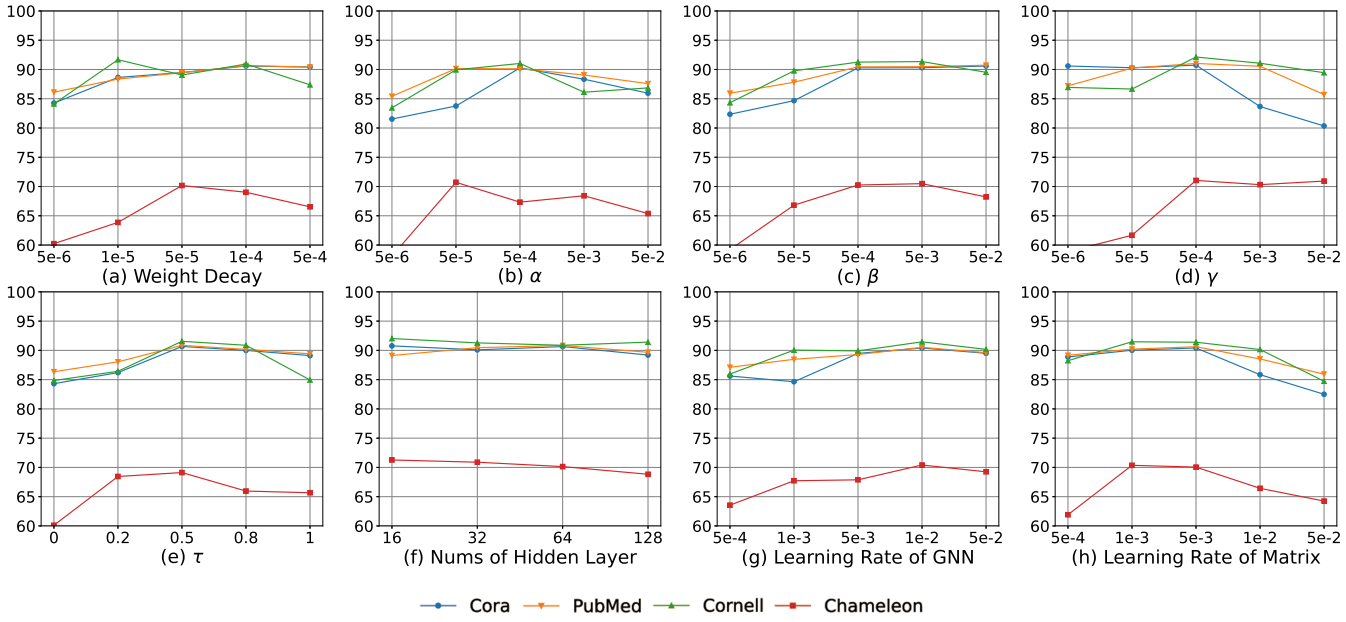

Fig. 2: Parameters sensitive analysis on Cora, Pubmed, Cornell and Chameleon.

[11] Yaqing Zheng, Hongzhi Zhang, Victor C. Lee, Yujia Zheng, Xiaofei Wang, and Shirui Pan, "Finding the missing-half: Graph complementary learning for homophily-prone and heterophily-prone graphs," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 42492–42505.

[12] Ellis Chien, Jingbo Peng, Pan Li, and Olgica Milenkovic, "Adaptive universal generalized pagerank graph neural network," in *International Conference on Learning Representations*, 2021.

[13] Meng He, Zhichun Wei, Zhiqiang Huang, and Hu Xu, "Bernnet: Learning arbitrary graph spectral filters via bernstein approximation," in *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021, pp. 14239–14251.

[14] Sitao Luan, Chenqing Hua, Qincheng Lu, Jian Zhu, Ming Zhao, Shiji Zhang, Xiao-Wen Chang, and Doina Precup, "Revisiting heterophily for graph neural networks," in *Proceedings of the 36th Conference on Neural Information Processing Systems*, 2022.

[15] Kun Huang, Yugan Wang, Meng Li, and Pietro Liò, "How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 20310–20330.

[16] Biao Li, Enqi Pan, and Zhihua Kang, "Pc-conv: Unifying homophily and heterophily with two-fold filtering," in *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, 2024, pp. 13437–13445.