HW-GNN: Homophily-Aware Gaussian-Window Constrained Graph Spectral Network for Social Network Bot Detection

Anonymous ACL submission

Abstract

Social bots are increasingly polluting online platforms by spreading misinformation and engaging in coordinated manipulation, posing severe threats to cybersecurity. Graph Neural Networks (GNNs) have become mainstream for social bot detection due to their ability to integrate structural and attribute features, with spectral-based approaches demonstrating particular efficacy due to discriminative patterns in the spectral domain. However, current spectral GNN methods face two limitations: (1) their broad-spectrum fitting mechanisms degrade the focus on bot-specific spectral features, and (2) certain domain knowledge valuable for bot detection, *e.g.*, low homophily correlates with high-frequency features, has not been fully incorporated into existing methods.

To address these challenges, we propose HW-GNN, a novel homophily-aware graph spectral network with Gaussian window constraints. Our framework introduces two key innovations: (i) a Gaussian-window constrained spectral network that employs learnable Gaussian windows to highlight bot-related spectral features, and (ii) a homophily-aware adaptation mechanism that injects domain knowledge between homophily ratios and frequency features into the Gaussian window optimization process. Through extensive experimentation on multiple benchmark datasets, we demonstrate that HW-GNN achieves state-of-the-art bot detection performance, outperforming existing methods with an average improvement of 4.6% in F1-score, while exhibiting strong plug-in compatibility with existing spectral GNNs.

1 Introduction

Social bots have become a pervasive threat in online social networks (Abulaish and Fazil, 2020), deceiving users by disseminating false information, amplifying harmful content, and coordinating manipulative campaigns (Domalewska, 2021; Ferrara, 2017) that can influence elections and undermine trust in digital platforms (Hajli et al., 2022). The detection of social bots is particularly challenging due to their rapidly evolving sophistication. Modern social bots are capable of closely mimicking human behavior, including realistic posting schedules (Khaund et al., 2021), tweet content generation (Grimme et al., 2022), and user profile imitation (Heidari et al., 2020). Moreover, by adaptively adjusting connection strategies (Ghiurău and Popescu, 2024), bots can further obscure identities, making it increasingly difficult to distinguish them from genuine users.

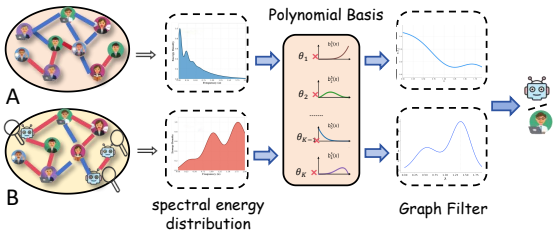


Figure 1: Illustration of homophily and spectral graph filtering in social bot detection. Homophilic community A: nodes with same labels form dense connections and concentrate spectral energy in low frequencies. Heterophilic community B: cross-label connections create dispersed patterns, shifting spectral energy toward higher frequencies. The spectral representations are then processed using polynomial basis functions to construct graph filters for bot detection.

Graph neural networks (GNNs) have become the mainstream approach for social bot detection by effectively leveraging both structural and attribute information (Liu et al., 2023a). Existing GNN-based methods can be broadly divided into spatial and spectral approaches. Spatial methods focus on message passing and aggregation in the node domain (Cai et al., 2021), capturing local neighborhood information through iterative feature propagation. These methods typically employ graph convolution operations that aggregate information from neighboring nodes, enabling the model to

learn node representations based on local structural context. While representative works such as BotRGCN (Feng et al., 2021b), BotMoE (Liu et al., 2023b), and H2GCN (Shao et al., 2024) have made significant progress in this area and demonstrated strong performance in various social network scenarios, spatial approaches are inherently dependent on the topological structure of graphs, where different social networks feature distinct structural patterns and varying sparsity levels that may limit generalization (Luan et al., 2023; Huo et al., 2023).

Bot detection can be viewed as an anomaly detection problem (Rafique et al., 2024), making graph spectral methods particularly promising for this task. As illustrated in Figure 1, these methods leverage spectral graph theory and polynomial approximations to transform node features into the spectral domain, where polynomial-based filters are applied to capture global structural patterns and anomalies that may not be apparent in the spatial domain. A representative method is BWGNN (Tang et al., 2022), which uses Beta polynomial basis functions to capture spectral features. Through mathematical derivation and empirical visualization, BWGNN demonstrates the relationship between heterophily ratio and spectral energy distribution on graphs with varying heterophily levels, showing that higher heterophily leads to spectral energy concentration in higher frequency bands. Other notable spectral methods include BernNet (He et al., 2021), JacobiConv (Wang and Zhang, 2022), and PolyGCL (Chen et al., 2024), achieving good performance in bot detection scenarios.

While spectral GNN methods are capable of fitting diverse spectral patterns through flexible polynomial filters (Zeng et al., 2023; Guo et al., 2023) and demonstrating strong performance in social bot detection tasks, they may be less sensitive to local bot-specific spectral anomalies that are crucial for effective bot detection. Most existing approaches use global polynomial filters, which fit the entire spectral domain by optimizing polynomial bases and their coefficients with respect to a loss defined over the full spectral domain. In practice, this global fitting may sometimes lose accuracy in critical bands where bot signals are most prominent, as anomalous behaviors tend to exhibit abrupt changes (Yu et al., 2019) that are not easily captured by filters optimized for overall spectral approximation.

In addition, such a valuable relationship between graph homophily and spectral energy has not been fully leveraged by the spectral GNN methods for improved bot detection performance. As illustrated in Figure 1, homophily characteristics directly influence spectral energy distribution (Tang et al., 2022; Gao et al., 2023; Xu et al., 2024). This relationship provides valuable domain knowledge for determining spectral focus regions in bot detection, but is still underutilized in current spectral GNN approaches.

To address limitations above, we propose HW-GNN, a homophily-aware Gaussian-window constrained graph spectral network for bot detection. The contributions of our method are summarized as follows:

- **Gaussian-Window constrained Spectral Network:** We introduce a Gaussian-window constrained spectral network that enables precise local spectral-domain fitting by employing learnable Gaussian functions as soft polynomial basis masks. This approach uses multiple Gaussian windows to extract features focusing on different frequency bands for comprehensive anomaly capture, providing precise local spectral filtering and plug-in compatibility with existing spectral GNNs (e.g., BWGNN, BernNet).

- **Homophily-Aware Adaptation Mechanism:** We innovatively inject domain knowledge by measuring the graph's homophily ratio and using a frequency distribution loss to adaptively guide the Gaussian window parameters, ensuring that spectral focus aligns with homophily-driven frequency preferences. This mechanism incorporates domain knowledge from bot detection to adaptively guide spectral focus.

Extensive experiments on five widely used real-world benchmarks demonstrate that HW-GNN achieves an average improvement of 4.6% in F1-score over state-of-the-art baselines, validating the effectiveness and plug-in compatibility of our approach.

2 Related Work

In recent years, graph-based approaches have emerged as a promising direction for social bot detection, as such approaches enable the integration of relational and user profile information within social networks. In this section, we review related work on graph-based bot detection, focusing on

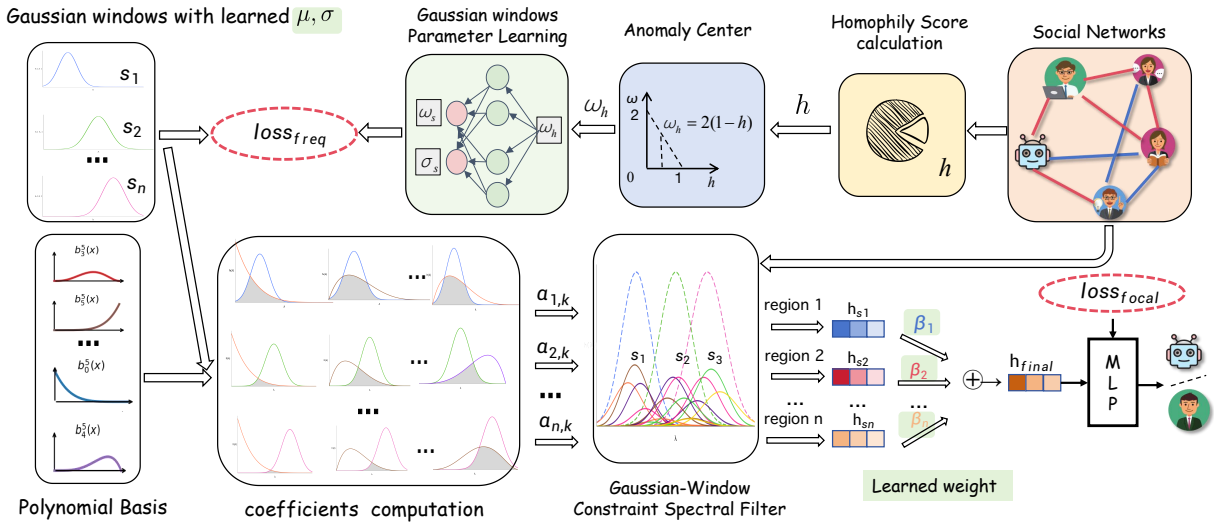


Figure 2: HW-GNN Framework: homophily-aware Gaussian-window constrained graph spectral network for social bot detection. The framework employs learnable Gaussian windows to modulate the weights of polynomial basis functions, enabling focused spectral analysis on bot-discriminative frequency bands. The homophily-aware adaptation mechanism injects domain knowledge between homophily ratios and frequency features to guide window parameter learning.

GNNs for homophilic and heterophilic graphs, spatial GNN methods, and spectral GNN methods.

Early GNN models such as GCN (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017), and SGC (Wu et al., 2019) are primarily designed under the homophily assumption, where nodes with similar labels are densely connected. However, many real-world social networks exhibit heterophilic or mixed connectivity patterns, where nodes with different labels are frequently connected. To address this challenge, a series of methods have been developed to better handle heterophilic graphs, such as CPGNN (Zhu et al., 2021), FAGCN (Bi et al., 2022), and H2GCN (Shao et al., 2024). These approaches relax the homophily assumption and enable more effective information aggregation in heterophilic settings, improving the robustness and generalization of GNNs in complex scenarios.

Spatial GNN methods leverage network topology and neighborhood aggregation to capture structural patterns, achieving strong performance in bot detection. Meanwhile, methods such as RGT (Feng et al., 2022a) and SlimG (Yoo et al., 2023) further improve robustness and interpretability. For bot detection, specialized spatial GNNs have been proposed, including SeBot (Yang et al., 2024), BSG4Bot (Miao et al., 2025), and BotBR (Lin and Zhou, 2025), which introduce relational modeling, balanced feature fusion, reliability-enhanced learning, or multi-view contrastive learning to better capture diverse bot behaviors. While spatial GNNs are effective at modeling local patterns, their effectiveness is constrained by the graph topology, as diverse social networks exhibit varying structural characteristics and connectivity densities that restrict their adaptability (Luan et al., 2023; Tang et al., 2020).

Spectral GNN methods operate in the spectral domain to capture global frequency-specific patterns. Representative approaches include ChebNet (Defferrard et al., 2016), BernNet (He et al., 2021), GPR-GNN (Chien et al., 2020), BWGNN (Tang et al., 2022), and JacobiConv (Wang and Zhang, 2022), which utilize various polynomial bases for spectral filtering. More recent works such as PolyGCL (Chen et al., 2024) that combines polynomial filters with self-supervised learning, and TFGNN (Li et al., 2025) further enhance the expressiveness and adaptability of spectral filters. While these methods can fit a wide range of frequency responses, most existing spectral GNNs may lack sensitivity to local spectral anomalies and often do not explicitly incorporate domain knowledge, which is crucial for distinguishing bots from genuine users in complex networks.

Different from prior work, HW-GNN introduces learnable Gaussian windows to locally focus spectral filtering on critical frequency bands. Furthermore, it adapts Gaussian window's parameters based on the graph's homophily ratio to leverage

bot detection domain knowledge, ensuring precise anomaly capture aligned with structural homophily.

3 Methodology

The framework of HW-GNN is shown in Figure 2. It consists of two major components: Gaussian-Window constrained Spectral Network and Homophily-Aware Adaptation Mechanism. First, the Gaussian-Window constrained Spectral Network employs learnable Gaussian windows with polynomial approximation to concentrate filtering on discriminative frequency bands, enabling precise spectral analysis for bot detection. Subsequently, the Homophily-Aware Adaptation Mechanism measures graph homophily ratios and adaptively tunes window parameters to ensure spectral focus aligns with anomaly characteristics, achieving superior performance through valuable domain knowledge injection.

3.1 Problem Formulation

We represent the social network as a heterogeneous attributed graph $G = (V, E, \mathcal{X}, \mathcal{R})$, where V is the set of users, $\mathcal{X} \in \mathbb{R}^{n \times d_0}$ their feature vectors, and $\mathcal{R}$ the relation types (e.g., "follows," "mentions," "retweets") with edges $E_r \subseteq V \times V$. While our framework supports heterogeneous graphs and can integrate information from different relation types, we focus on single-relation GNNs in the following, and multiple relationships can be supported through attention-based methods. A subset $V_L \subseteq V$ has binary labels $y_i \in \{0, 1\}$ indicating genuine and bot users. We aim to learn

$$f : G \longmapsto [0, 1]^n \tag{1}$$

to predict each node's bot probability $\hat{y}_i$.

To adapt to mixed homophily, we compute the graph-level homophily ratio (Ma et al., 2022), and assume that the target frequency band is inversely correlated with the homophily ratio, *i.e.*, lower homophily leads to a preference for higher spectral components.

$$h = \frac{\left|\{(u, v) \in E : y_u = y_v\}\right|}{|E|} \tag{2}$$

where $E = \bigcup_r E_r$. The scalar h can be used to guide the positioning of Gaussian windows in our spectral filters.

3.2 Gaussian-Window constrained Spectral Network

The core innovation of our approach lies in introducing learnable Gaussian windows that modulate polynomial basis coefficients to achieve local spectral focusing on bot-discriminative frequency bands, enabling the model to extract and combine the most informative spectral features for distinguishing bots from genuine users.

3.2.1 Gaussian Window Spectral Focusing Strategy

While spectral GNNs with global filters can capture various spectral patterns, they may degrade focus to detect abrupt frequency changes associated with anomaly features. To address this issue, we employ S learnable Gaussian windows that focus on specific frequency bands for comprehensive bot detection:

$$G_s(\lambda) = \exp\left(-\tfrac{(\lambda-\omega_s)^2}{2\sigma_s^2}\right), \quad s = 1, \ldots, S \tag{3}$$

Here $\lambda \in [0, 2]$ is a Laplacian eigenvalue, ω_s the center frequency, and σ_s the window's bandwidth. Each Gaussian window $G_s(\lambda)$ defines a localized spectral filter that focuses on specific frequency regions. By learning $\{\omega_s, \sigma_s\}$, the model adaptively adjusts the center frequencies and bandwidths of Gaussian windows to capture information from different spectral regions and the combination of multiple such filters enables comprehensive spectral analysis.

Direct spectral filtering requires eigendecomposition of the Laplacian matrix with computational complexity $\mathcal{O}(n^3)$, which is prohibitive for large graphs. Therefore, for our Gaussian-window constrained filters $g_s(\lambda)$, we use polynomial basis functions $\{P_k^{(K)}(\lambda)\}_{k=0}^K$ of order K to approximate the filter response:

$$g_s(\lambda) \approx \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\lambda) \tag{4}$$

The interaction coefficient $c_{s,k}$ reflects how much each polynomial basis function contributes to approximating the s-th Gaussian-Window constrained filter. For all polynomial basis functions $\{P_k^{(K)}(\lambda)\}_{k=0}^K$ and a given Gaussian window $G_s(\lambda)$, we compute the contribution weight of each basis function:

$$c_{s,k} = \int_0^2 G_s(\lambda) P_k^{(K)}(\lambda)\, d\lambda \tag{5}$$

These coefficients are determined by the overlap between the Gaussian window and the polynomial basis functions in spectral region. *The **greater** the overlap between a polynomial basis and the Gaussian window's frequency region, the **larger** the corresponding coefficient $c_{s,k}$, resulting in stronger emphasis on the corresponding spectral region.*

The theoretical proof that multi-Gaussian Window constrained polynomial fitting outperforms global fitting is detailed in Appendix Section A.2.

3.2.2 Gaussian-Window constrained Spectral Filter Implementation

With the coefficients obtained from the Gaussian window analysis, we implement the spectral filtering by integrating multiple Gaussian-windowed filters.

Starting from the spectral filter with integration coefficients, we derive the complete implementation through the following steps:

$$g_s(\mathbf{L}) = \int_0^2 G_s(\lambda)\, d\mathbf{E}(\lambda) \tag{6}$$

$$\approx \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\mathbf{L}) \tag{7}$$

$$\mathbf{Z}_s = g_s(\mathbf{L})\mathbf{X}\mathbf{W}_s \tag{8}$$

$$= \left(\sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\mathbf{L}) \right) \mathbf{X}\mathbf{W}_s \tag{9}$$

$$= \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\mathbf{L})\mathbf{X}\mathbf{W}_s \tag{10}$$

where $\mathbf{E}(\lambda)$ is the spectral measure, $c_{s,k}$ are the coefficients from Eq. (5) that reflect the overlap between Gaussian windows and polynomial basis functions, $P_k^{(K)}(\mathbf{L})$ are the polynomial basis matrices applied to the Laplacian, $\mathbf{X}$ is the input feature matrix, and $\mathbf{W}_s \in \mathbb{R}^{d \times d'}$ is the learnable weight matrix for the s-th window. This strategy does not change the essential nature of graph convolution, with theoretical proof provided in Appendix Section A.1.

The multi-window output combines all spectral features through learned weights:

$$\mathbf{H} = \sum_{s=1}^{S} w_s \mathbf{Z}_s \tag{11}$$

$$= \sum_{s=1}^{S} w_s \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\mathbf{L})\mathbf{X}\mathbf{W}_s \tag{12}$$

where w_s are learned importance weights computed through a softmax function:

$$w_s = \frac{\exp(\beta_s)}{\sum_{j=1}^{S} \exp(\beta_j)} \tag{13}$$

with learnable parameters β_s, ensuring $\sum_{s=1}^{S} w_s = 1$.

3.3 Homophily-Aware Adaptation Mechanism

The center frequencies and bandwidths of Gaussian windows should adapt to the structural characteristics of different social networks. Our solution leverages the relationship between graph homophily and spectral energy distribution to guide the learning of optimal window parameters.

3.3.1 Gaussian Window Parameter Learning

For each edge type e with homophily ratio h_e, we set the expected anomaly center frequency as

$$\bar{\omega}(h_e) = 2\,(1 - h_e) \tag{14}$$

This mapping reflects the spectral theory insight that high homophily ($h_e \to 1$) concentrates energy in low frequencies ($\bar{\omega} \to 0$), while low homophily ($h_e \to 0$) shifts energy toward high frequencies ($\bar{\omega} \to 2$).

While the above relationship provides a theoretical mapping between homophily ratios and center frequencies, this mapping is empirical and should not strictly constrain the learned frequencies. We allow some deviation around the target frequencies to provide flexibility in learning. To inject this homophily-driven domain knowledge into the spectral filtering process, we employ MLP networks to learn the center frequencies and bandwidths of the Gaussian windows. Specifically, the MLPs take the homophily-guided target frequency $\bar{\omega}(h_e)$ as input and output the parameters for each window:

$$\omega_s = \text{MLP}_\omega(\bar{\omega}(h_e), s) \tag{15}$$

$$\sigma_s = \text{MLP}_\sigma(\bar{\omega}(h_e), s) \tag{16}$$

where $\text{MLP}_\omega : \mathbb{R}^2 \to [s^-, s^+]$ and $\text{MLP}_\sigma : \mathbb{R}^2 \to \mathbb{R}^+$ are multi-layer perceptrons. The center frequencies ω_s are initialized as S equally-spaced points in the spectral domain $[0, 2]$, which provides comprehensive coverage of the entire frequency range.

During training, the MLPs learn adaptive offsets from these initial positions toward the homophily-guided target $\bar{\omega}(h_e)$, enabling the model to adjust

spectral focus based on graph structural properties. To maintain local spectral focus and prevent excessive deviation, the MLP output is clipped to a constrained range $[s^-, s^+]$ around each window's initial position, ensuring that each Gaussian window remains concentrated on its designated local frequency region. By explicitly feeding the homophily-guided target frequency into the MLPs, we ensure that the parameterization of each Gaussian window is directly influenced by the structural properties of the graph.

Furthermore, the entire process is trained end-to-end with a frequency distribution loss, which continuously encourages the MLPs to produce window parameters that remain closely aligned with the homophily-driven spectral targets. This design enables the model to flexibly adapt to diverse network structures while still preserving the theoretical connection between homophily and spectral focus.

To further reinforce this alignment, we introduce a frequency distribution loss that anchors the learned window centers to positions determined by the homophily ratio:

$$\mathcal{L}_{\text{freq}} = \frac{1}{C}\sum_{c=1}^{C}\left(\hat{\omega}^{(c)} - \bar{\omega}(h_e)\right)^2 \tag{17}$$

where $\hat{\omega}^{(c)}$ is the learned center frequency in the convolution block c and C is the total number of blocks.

The Gaussian window parameter learning mechanism thus injects domain knowledge by measuring the graph's homophily ratio and uses a frequency distribution loss to guide the Gaussian window parameters, ensuring that spectral focus aligns with homophily-driven frequency preferences.

3.4 Multi-Layer Architecture

We stack multiple Gaussian-windowed convolution layers with residual connections:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\mathbf{H}^{(\ell)} + \text{HW-Conv}(\mathbf{H}^{(\ell)})\right) \tag{18}$$

where $\mathbf{H}^{(\ell)}$ denotes features at layer ℓ, $\sigma(\cdot)$ is the activation function, HW-Conv denotes the Gaussian-window constrained spectral network, and the residual connection helps mitigate over-smoothing.

3.5 Overall Loss Function

The complete loss function incorporates the classification loss and frequency distribution loss:

$$\mathcal{L} = \mathcal{L}_{\text{focal}} + \lambda_f \mathcal{L}_{\text{freq}} \tag{19}$$

where $\mathcal{L}_{\text{focal}}$ is the Focal Loss for addressing class imbalance:

$$\mathcal{L}_{\text{focal}} = -\frac{1}{|V_L|}\sum_{i\in V_L}\alpha_i(1-\hat{y}_i)^\gamma \log(\hat{y}_i) \tag{20}$$

where $|V_L|$ is the number of training nodes, $\hat{y}_i$ is the predicted probability for the true class, α_i is a balancing factor, and γ is the focusing parameter. $\lambda_f < 1$ is a weighting parameter that balances classification accuracy with frequency distribution regularization.

3.6 Computational Complexity Analysis

Let S be the number of Gaussian windows, K be the polynomial order, $|E|$ be the number of edges, d and d' be the input and output feature dimensions, and d_{MLP} be the number of parameters in the MLPs for window parameter learning.

Our method consists of two main computational steps. For step (i), the Gaussian window parameter learning and polynomial coefficient computation requires $\mathcal{O}(S \cdot d_{\text{MLP}} + S \cdot K)$ operations. For step (ii), the multi-window filtering requires $\mathcal{O}(S \cdot K \cdot |E| \cdot d \cdot d')$ operations.

The overall computational complexity is $\mathcal{O}(S \cdot K \cdot |E| \cdot d \cdot d' + S \cdot d_{\text{MLP}} + S \cdot K)$. The dominant term $S \cdot K \cdot |E| \cdot d \cdot d'$ scales linearly with the graph size and feature dimensions. While HW-GNN introduces additional complexity for Gaussian window parameter learning compared to baseline polynomial methods, this overhead only affects the less computationally intensive coefficient computation step, ensuring scalability for large-scale social networks.

4 Experiments

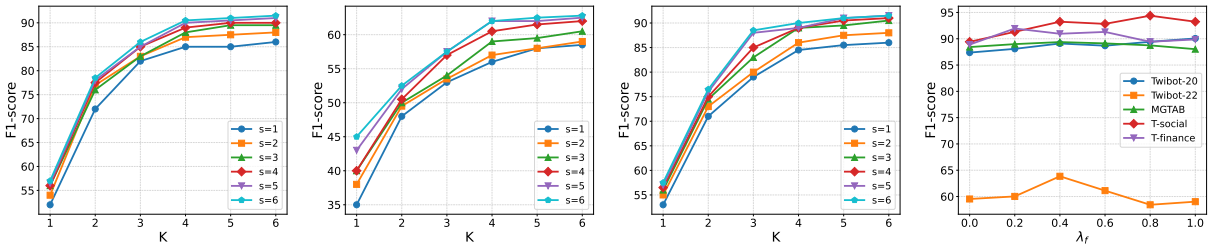
4.1 Experimental Setup

4.1.1 Datasets

We evaluate HW-GNN on five widely-adopted benchmarks spanning different scales and homophily patterns. TwiBot-20 (Feng et al., 2021a) and TwiBot-22 (Feng et al., 2022b) are comprehensive Twitter datasets with mixed relations, with the latter scaling to 1 million users. MGTAB (Shi et al., 2023) features heterogeneous connections with seven relation types. T-Social and T-Finance (Tang et al., 2022) represent massive homophilic and financial networks, respectively. Detailed statistics are provided in Appendix Section B.1.

Model	Twibot-20		Twibot-22		MGTAB		T-social		T-finance	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
MLP	83.89±1.10	81.71±0.80	79.01±0.70	53.81±0.30	84.88±0.40	84.67±0.30	71.52±0.45	48.35±0.62	78.85±0.55	70.57±0.58
GCN (2017)	77.52±0.20	80.85±0.40	78.41±0.40	54.91±0.40	83.65±0.20	84.02±0.60	75.87±0.35	59.88±0.55	79.03±0.42	70.74±0.45
GAT (2018)	83.33±0.30	81.26±0.70	79.54±0.30	55.83±0.90	84.45±0.30	83.69±0.40	76.82±0.38	69.01±0.65	77.21±0.48	53.86±0.52
BotRGCN (2021)	85.86±0.80	87.33±0.70	78.56±0.10	57.52±1.20	89.69±1.10	86.02±1.20	77.65±0.52	68.13±0.75	75.98±0.55	52.14±0.65
RGT (2022)	86.67±0.30	88.22±0.10	76.44±0.20	43.02±0.70	89.76±0.40	86.59±0.80	77.98±0.45	71.23±0.68	86.27±0.42	87.12±0.45
GPR-GNN (2020)	87.47±0.80	88.84±1.20	78.64±0.60	57.66±0.80	90.32±0.40	87.46±1.10	75.13±0.65	59.76±0.85	84.78±0.55	85.07±0.62
SlimG (2023)	86.55±0.30	87.97±0.50	74.76±0.30	44.27±1.60	88.13±0.30	84.45±0.70	74.76±0.42	58.73±0.85	82.21±0.45	84.54±0.52
H2GCN (2024)	88.23±0.60	89.14±0.70	77.64±0.40	57.23±1.30	90.56±0.50	87.72±0.30	–	–	87.93±0.48	88.24±0.52
SeBot (2024)	87.24±0.10	88.74±0.13	78.15±0.21	58.34±0.35	90.46±1.44	82.12±2.42	67.54±1.12	65.89±1.05	86.45±0.18	86.12±0.22
BSG4Bot (2025)	89.15±0.40	89.89±0.20	79.53±0.20	59.42±1.30	91.75±0.70	88.53±1.50	68.21±0.55	66.15±0.85	87.23±0.45	86.82±0.48
BotBR (2025)	87.23±0.08	88.91±0.06	78.92±0.15	59.15±0.18	91.81±0.30	86.39±1.19	69.45±0.85	67.23±0.92	87.83±0.15	87.05±0.14
BWGNN (2022)	85.23±0.55	87.62±0.65	58.21±0.45	59.78±0.85	89.54±0.42	87.13±0.58	81.43±0.52	83.98±0.65	85.76±0.48	86.87±0.52
BernNet (2021)	87.78±0.45	88.43±0.55	78.93±0.35	59.98±0.75	79.56±0.48	88.43±0.62	76.56±0.55	84.13±0.68	86.16±0.45	87.63±0.52
JacobiConv (2022)	84.58±0.52	88.01±0.62	59.94±0.42	60.03±0.82	90.72±0.45	87.96±0.55	61.19±0.58	75.51±0.72	82.21±0.52	86.91±0.55
TFGNN (2025)	88.98±0.48	89.13±0.52	59.89±0.38	59.71±0.78	89.98±0.42	88.14±0.52	81.29±0.55	84.07±0.65	87.89±0.48	88.05±0.52
HW-Beta	87.23±0.12	88.16±0.15	77.54±0.08	60.98±0.25	91.43±0.18	88.93±0.22	89.50±0.15	91.16±0.18	88.16±0.12	89.59±0.15
HW-Jacobi	88.80±0.10	88.50±0.12	78.84±0.09	60.31±0.22	89.45±0.15	**89.87±0.18**	**96.71±0.12**	**94.47±0.15**	**90.46±0.10**	91.67±0.12
HW-BernStein	**90.37±0.08**	**91.51±0.10**	**80.73±0.06**	**61.95±0.18**	**92.83±0.12**	89.37±0.15	96.53±0.10	94.39±0.12	88.91±0.08	**91.95±0.10**
Improvement	**+1.37%**	**+1.80%**	**+1.50%**	**+3.20%**	**+1.11%**	**+1.51%**	**+18.76%**	**+12.29%**	**+2.89%**	**+4.20%**

Table 1: Accuracy and F1-score of Competitors on the Five Benchmarks.



(a) Twibot-20 (b) Twibot-22 (c) MGTAB (d) Loss weight λ_f

Figure 3: Sensitivity analysis of key parameters in HW-GNN. (a)–(c): F1-score on Twibot20, Twibot22, and MGTAB datasets, respectively, with varying polynomial order K and different numbers of Gaussian windows S (each line represents a different S). (d): F1-score variation with respect to the frequency distribution loss weight λ_f.

4.1.2 Baselines

We compare HW-GNN against representative baselines spanning three categories: content-based method MLP; spatial GNNs including GCN (Kipf and Welling, 2016), GAT (Velickovic et al., 2017), BotRGCN (Feng et al., 2021b), RGT (Feng et al., 2022a), SlimG (Yoo et al., 2023), H2GCN (Shao et al., 2024), SeBot (Yang et al., 2024), BSG4Bot (Miao et al., 2025), and BotBR (Lin and Zhou, 2025); and spectral GNNs including GPR-GNN (Chien et al., 2020), BernNet (He et al., 2021), BWGNN (Tang et al., 2022), JacobiConv (Wang and Zhang, 2022), and TFGNN (Li et al., 2025).

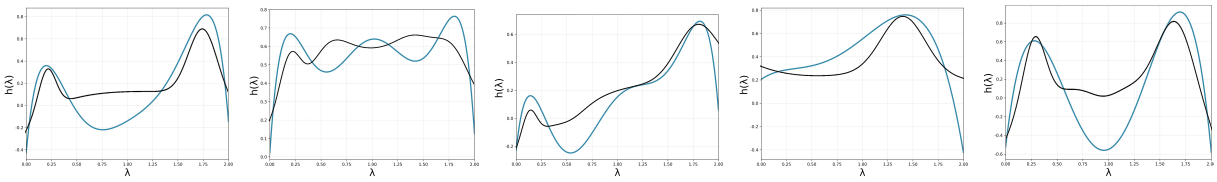
4.1.3 Implementation Details

HW-GNN is implemented in PyTorch and DGL. All experiments run on a server with 256GB RAM, dual Intel Xeon Silver CPUs @2.4GHz, and an NVIDIA RTX A6000 GPU (48GB). To avoid overfitting, we use early stopping based on validation Macro-F1.

4.2 Performance on different Baselines

The experimental results are presented in Table 1. We compare HW-GNN against all baselines across the five benchmarks, reporting accuracy and F1-score results.

HW-GNN achieves superior performance across all benchmarks, demonstrating the effectiveness of our homophily-aware Gaussian-Window constrained spectral network. On TwiBot-20, our method outperforms the best baseline BSG4Bot by 1.8% F1-score, showcasing the advantage of our Gaussian window mechanism in capturing bot-specific spectral features. For the challenging heterophilic TwiBot-22 dataset, we achieve 3.2% F1 improvement over the second-best method JacobiConv, highlighting the effectiveness of our homophily-aware adaptation in handling complex heterophilic structures. Our method achieves the most significant improvements on large-scale datasets. T-Social achieves 12.3% F1 improvement over BernNet and T-Finance shows 4.2% F1 improvement over H2GCN, demonstrating the supe-



(a) TwiBot-20 (b) TwiBot-22 (c) MGTAB (d) T-Finance (e) T-social

Figure 4: Learned spectral filter $h(\lambda)$ for different datasets by HW-GNN (blue) and original Bernstein polynomial (black).

Model	Twibot-20		Twibot-22		MGTAB		T-social		T-finance	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
HW-GNN (Full)	**90.37±0.08**	**91.51±0.10**	**80.73±0.06**	**61.85±0.18**	**92.83±0.12**	**89.37±0.15**	**96.53±0.10**	**94.39±0.12**	**88.91±0.08**	**91.95±0.10**
w/o Gaussian Window	87.78±0.45	88.43±0.55	78.93±0.35	59.98±0.75	79.56±0.48	88.43±0.62	76.56±0.55	84.13±0.68	86.16±0.45	87.63±0.52
w/o Homophily-Aware	88.73±0.25	89.42±0.32	79.25±0.22	60.39±0.45	91.69±0.28	88.78±0.35	94.59±0.25	93.67±0.32	88.98±0.22	89.23±0.28
w/o Multi-band(S=1)	86.81±0.35	87.78±0.42	75.23±0.28	56.42±0.55	86.24±0.32	82.32±0.45	92.83±0.38	90.27±0.48	86.95±0.35	88.43±0.42

Table 2: Ablation Study Results on the five Datasets.

rior capability of our approach in processing large-scale social networks. On MGTAB, our method surpasses BotBR by 1.1% accuracy, further validating the robustness of our HW-GNN across diverse graph structures.

Notably, our HW-GNN demonstrates strong plug-in compatibility with existing spectral architectures and consistently outperforms their corresponding standalone polynomial GNNs. The results validate that our Gaussian-Window constrained spectral network can be effectively integrated with different polynomial bases (Beta, Jacobi, and Bernstein), with each HW variant showing significant improvements over the original polynomial methods, highlighting the broad applicability of our approach. The learned spectral filter responses $h(\lambda)$ comparing HW-GNN with original spectral methods are visualized in Figure 4.

4.3 Ablation Studies

We perform ablation experiments using the Bernstein polynomial basis on five datasets (Table 2). The results demonstrate that all components contribute significantly to the overall performance. Removing the Gaussian-Window component causes F1 to drop ranging from 0.94% to 10.26%, confirming that the Gaussian-Window constrained spectral network is essential for effective spectral focusing. The homophily-aware adaptation mechanism also shows consistent importance, with F1 declines ranging from 0.59% to 2.72% when removed, validating the effectiveness of homophily domain knowledge injection for bot detection.

Moreover, replacing multiple Gaussian windows with a single global filter results in significant performance degradation, with F1 drops ranging from 2.28% to 7.05%. This demonstrates the critical necessity of multiple Gaussian windows for capturing diverse bot behavioral patterns across different frequency bands, particularly in networks where bots exhibit varied spectral features.

4.4 Sensitivity Analysis of Key Parameters

We analyze parameter sensitivity in Figure 3. Performance generally improves with increased Gaussian windows S and polynomial order K, plateauing around S values of 4 to 6 and K values of 3 to 4. We select $S = 5$ and $K = 4$ for the optimal accuracy-efficiency trade-off. Regarding loss weight λ_f, most datasets such as TwiBot-22 peak at lower values ranging from 0.2 to 0.4, while others like TwiBot-20 benefit from stronger homophily guidance with higher weights.

Furthermore, robustness tests varying training ratios from 10% to 80% show HW-GNN consistently outperforms baselines, especially in low-resource settings. This confirms our mechanism effectively captures generalized spectral patterns with limited supervision.

5 Conclusion

In this paper, we addressed the key limitation of existing spectral GNNs for their insufficient ability to achieve precise local region focusing and inject valuable homophily domain knowledge into bot detection. By employing Gaussian-window constrained spectral network and incorporating homophily-aware adaptation mechanism, HW-GNN achieves superior performance across diverse social bot detection benchmarks.

Limitations

Our homophily-aware mechanism assumes a general correlation between homophily and spectral energy distribution, which might vary in specific adversarial settings where bots deliberately manipulate their connections to mimic human behavior. Although our method shows improved robustness, future work should investigate more advanced mechanisms against such sophisticated adversarial attacks. Additionally, while our Gaussian window strategy effectively captures local spectral patterns, extending it to dynamic graphs with temporally evolving spectral characteristics remains a direction for future exploration.

Ethical considerations

This work utilizes publicly available social network datasets for bot detection research. We strictly adhere to the privacy policies of the respective platforms and the datasets' terms of use. No personal identifiable information (PII) beyond what is publicly available in the datasets was processed or inferred. We acknowledge the potential dual-use of bot detection technology and emphasize that our method is intended for identifying automated accounts to improve platform integrity, not for suppressing legitimate user activities.

References

Muhammad Abulaish and Mohd Fazil. 2020. Socialbots: Impacts, threat-dimensions, and defense challenges. *IEEE Technology and Society Magazine*, 39(3):52–61.

Xia-an Bi, Wenyan Zhou, Sheng Luo, Yuhua Mao, Xi Hu, Bin Zeng, and Luyun Xu. 2022. Feature aggregation graph convolutional network based on imaging genetic data for diagnosis and pathogeny identification of alzheimer's disease. *Briefings in Bioinformatics*, 23(3):bbac137.

Shaofei Cai, Liang Li, Jincan Deng, Beichen Zhang, Zheng-Jun Zha, Li Su, and Qingming Huang. 2021. Rethinking graph neural architecture search from message-passing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6657–6666.

Jingyu Chen, Runlin Lei, and Zhewei Wei. 2024. Polygcl: Graph contrastive learning via learnable spectral polynomial filters. In *The Twelfth International Conference on Learning Representations*.

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

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Dorota Domalewska. 2021. Disinformation and polarization in the online debate during the 2020 presidential election in poland. *Safety & Defense*, 7(1):14–24.

Richard J Duffin and Albert C Schaeffer. 1941. A refinement of an inequality of the brothers markoff. *Transactions of the American Mathematical Society*, 50(3):517–528.

Tamás Erdélyi. 1993. Remez-type inequalities and their applications. *Journal of computational and applied mathematics*, 47(2):167–209.

Shangbin Feng, Zhaoxuan Tan, Rui Li, and Minnan Luo. 2022a. Heterogeneity-aware twitter bot detection with relational graph transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3977–3985.

Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, and 1 others. 2022b. Twibot-22: Towards graph-based twitter bot detection. *Advances in Neural Information Processing Systems*, 35:35254–35269.

Shangbin Feng, Herun Wan, Ningnan Wang, Jundong Li, and Minnan Luo. 2021a. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4485–4494.

Shangbin Feng, Herun Wan, Ningnan Wang, and Minnan Luo. 2021b. Botrgcn: Twitter bot detection with relational graph convolutional networks. In *Proceedings of the 2021 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 236–239.

Emilio Ferrara. 2017. Disinformation and social bot operations in the run up to the 2017 french presidential election. *First Monday*.

Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. 2023. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *Proceedings of the ACM web conference 2023*, pages 1528–1538.

David Ghiurău and Daniela Elena Popescu. 2024. Distinguishing reality from ai: Approaches for detecting synthetic content. *Computers*, 14(1):1.

Christian Grimme, Janina Pohl, Stefano Cresci, Ralf Lüling, and Mike Preuss. 2022. New automation for social bots: from trivial behavior to ai-powered communication. In *Multidisciplinary International Symposium on Disinformation in Open Online Media*, pages 79–99. Springer.

Jingwei Guo, Kaizhu Huang, Xinping Yi, and Rui Zhang. 2023. Graph neural networks with diverse spectral filtering. In *Proceedings of the ACM web conference 2023*, pages 306–316.

Nick Hajli, Usman Saeed, Mina Tajvidi, and Farid Shirazi. 2022. Social bots and the spread of disinformation in social media: the challenges of artificial intelligence. *British Journal of Management*, 33(3):1238–1253.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Mingguo He, Zhewei Wei, Hongteng Xu, and 1 others. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34:14239–14251.

Maryam Heidari, James H Jones, and Ozlem Uzuner. 2020. Deep contextualized word embedding for text-based online user profiling to detect social bots on twitter. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 480–487. IEEE.

Cuiying Huo, Di Jin, Yawen Li, Dongxiao He, Yu-Bin Yang, and Lingfei Wu. 2023. T2-gnn: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4339–4346.

Dunham Jackson. 1930. *The theory of approximation*, volume 11. American Mathematical Soc.

Tuja Khaund, Baris Kirdemir, Nitin Agarwal, Huan Liu, and Fred Morstatter. 2021. Social bots and their coordination during online campaigns: a survey. *IEEE Transactions on Computational Social Systems*, 9(2):530–545.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Guoming Li, Jian Yang, Shangsong Liang, and Dongsheng Luo. 2025. Polynomial selection in spectral graph neural networks: An error-sum of function slices approach. In *Proceedings of the ACM on Web Conference 2025*, pages 3276–3287.

Qilong Lin and Jingya Zhou. 2025. Botbr: Social bot detection with balanced feature fusion and reliability-enhanced graph learning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 392–402.

Xingyu Liu, Juan Chen, and Quan Wen. 2023a. A survey on graph classification and link prediction based on gnn. *arXiv preprint arXiv:2307.00865*.

Yuhan Liu, Zhaoxuan Tan, Heng Wang, Shangbin Feng, Qinghua Zheng, and Minnan Luo. 2023b. Botmoe: Twitter bot detection with community-aware mixtures of modal-specific experts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 485–495.

Sitao Luan, Chenqing Hua, Minkai Xu, Qincheng Lu, Jiaqi Zhu, Xiao-Wen Chang, Jie Fu, Jure Leskovec, and Doina Precup. 2023. When do graph neural networks help with node classification? investigating the homophily principle on node distinguishability. *Advances in Neural Information Processing Systems*, 36:28748–28760.

Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. 2022. Is homophily a necessity for graph neural networks? In *International Conference on Learning Representations*.

Hao Miao, Zida Liu, and Jun Gao. 2025. Bsg4bot: Efficient bot detection based on biased heterogeneous subgraphs. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 1236–1249. IEEE Computer Society.

Saida Hafsa Rafique, Amira Abdallah, Nura Shifa Musa, and Thangavel Murugan. 2024. Machine learning and deep learning techniques for internet of things network anomaly detection—current research trends. *Sensors*, 24(6):1968.

Yiming Shao, Lintao Mao, Leixiong Ye, Jincheng Li, Ping Yang, Chengtao Ji, and Zizhao Wu. 2024. H2gcn: A hybrid hypergraph convolution network for skeleton-based action recognition. *Journal of King Saud University-Computer and Information Sciences*, 36(5):102072.

Shuhao Shi, Kai Qiao, Jian Chen, Shuai Yang, Jie Yang, Baojie Song, Linyuan Wang, and Bin Yan. 2023. Mgtab: A multi-relational graph-based twitter account detection benchmark. *arXiv preprint arXiv:2301.01123*.

Jianheng Tang, Jiajin Li, Ziqi Gao, and Jia Li. 2022. Rethinking graph neural networks for anomaly detection. In *International conference on machine learning*, pages 21076–21089. PMLR.

Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Yiqi Wang, Jiliang Tang, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Investigating and mitigating degree-related biases in graph convoltuional networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1435–1444.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, and 1 others. 2017. Graph attention networks. *stat*, 1050(20):10–48550.

Xiyuan Wang and Muhan Zhang. 2022. How powerful are spectral graph neural networks. In *International conference on machine learning*, pages 23341–23362. PMLR.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. Pmlr.

Fan Xu, Nan Wang, Hao Wu, Xuezhi Wen, Xibin Zhao, and Hai Wan. 2024. Revisiting graph-based fraud detection in sight of heterophily and spectrum. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 9214–9222.

Yingguang Yang, Qi Wu, Buyun He, Hao Peng, Renyu Yang, Zhifeng Hao, and Yong Liao. 2024. Sebot: Structural entropy guided multi-view contrastive learning for social bot detection. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3841–3852.

Jaemin Yoo, Meng-Chieh Lee, Shubhranshu Shekhar, and Christos Faloutsos. 2023. Less is more: Slimg for accurate, robust, and interpretable graph mining. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3128–3139.

Guang Yu, Zhiping Cai, Siqi Wang, Haiwen Chen, Fang Liu, and Anfeng Liu. 2019. Unsupervised online anomaly detection with parameter adaptation for kpi abrupt changes. *IEEE Transactions on Network and Service Management*, 17(3):1294–1308.

Zeyuan Zeng, Qinke Peng, Xu Mou, Ying Wang, and Ruimeng Li. 2023. Graph neural networks with high-order polynomial spectral filters. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):12590–12603.

Jiong Zhu, Ryan A Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K Ahmed, and Danai Koutra. 2021. Graph neural networks with heterophily. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11168–11176.

Appendix Overview

This appendix provides supplementary materials for the main paper. Section A presents the detailed theoretical derivation proving that our Gaussian-windowed approach preserves the fundamental properties of graph convolution while offering superior local approximation capabilities. Section B includes additional experimental results, such as detailed dataset statistics and computational efficiency analysis, which further validate the effectiveness and scalability of HW-GNN.

A Theoretical Analysis and Proofs

In this appendix, we provide the theoretical foundation for our HW-GNN: Homophily-Aware Gaussian-Window Constraint Graph spectral network approach. We present two key theoretical proofs: (1) a step-by-step derivation proving that introducing Gaussian windows does not violate graph convolution theory and maintains the essential computational framework, and (2) mathematical proof that multi-interval fitting is superior to global fitting for spectral graph neural networks.

A.1 Gaussian-Windowed Graph Convolution Derivation

In this section, we prove that introducing Gaussian window functions does not change the essential nature of graph convolution and does not introduce new computational complexity, since the integral coefficients are constants.

A.1.1 From Continuous Spectral Integration to Discrete Polynomial Implementation

We start from continuous spectral domain integration and derive step-by-step to discrete polynomial graph convolution implementation:

Step 1: Continuous Spectral Domain Filtering

$$\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{X} = \int_0^2 g(\lambda)\, d\mathbf{E}(\lambda)\mathbf{X} \tag{21}$$

where $\mathbf{E}(\lambda)$ is the spectral measure and $g(\lambda)$ is the filter function.

Step 2: Introducing Gaussian Window Functions For the s-th Gaussian window:

$$g_s(\lambda) = \exp\left(-\frac{(\lambda - \omega_{0,s})^2}{2\sigma_s^2}\right) \tag{22}$$

The windowed spectral filtering becomes:

$$\mathbf{Y}_s = \int_0^2 g_s(\lambda)\, d\mathbf{E}(\lambda)\mathbf{X} \tag{23}$$

Step 3: Polynomial Basis Approximation Take Bernstein basis functions (normalized to $[0, 2]$ interval) for concrete illustration:

$$P_k^{(K)}(\lambda) = \binom{K}{k}\left(\frac{\lambda}{2}\right)^k\left(1 - \frac{\lambda}{2}\right)^{K-k} \tag{24}$$

Step 4: Computing Integral Coefficients (Key Step) The integral coefficients between Gaussian windows and Bernstein basis functions:

$$c_{s,k} = \int_0^2 g_s(\lambda)P_k^{(K)}(\lambda)\, d\lambda \tag{25}$$

$$= \int_0^2 \exp\left(-\frac{(\lambda - \omega_{0,s})^2}{2\sigma_s^2}\right)\binom{K}{k}\left(\frac{\lambda}{2}\right)^k\left(1 - \frac{\lambda}{2}\right)^{K-k} d\lambda \tag{26}$$

Key Observation: $c_{s,k}$ are constants

$c_{s,k}$ are constants independent of graph structure, depending only on window parameters $\omega_{0,s}, \sigma_s$ and polynomial order K

Step 5: Polynomial Approximation

$$g_s(\lambda) \approx \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\lambda) \tag{27}$$

Step 6: Matrix Form of Polynomial Approximation

$$g_s(\mathbf{L}) \approx \sum_{k=0}^{K} c_{s,k} P_k^{(K)}(\mathbf{L}) \tag{28}$$

Step 7: Expanding Bernstein Polynomials in Matrix Form For $P_k^{(K)}(\mathbf{L}) = \binom{K}{k} \left(\frac{\mathbf{L}}{2}\right)^k \left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^{K-k}$, we have:

$$P_k^{(K)}(\mathbf{L}) = \binom{K}{k} \left(\frac{1}{2}\right)^k \mathbf{L}^k \sum_{j=0}^{K-k} \binom{K-k}{j} (-1)^j \left(\frac{1}{2}\right)^j \mathbf{L}^j \tag{29}$$

$$= \binom{K}{k} \left(\frac{1}{2}\right)^k \sum_{j=0}^{K-k} \binom{K-k}{j} (-1)^j \left(\frac{1}{2}\right)^j \mathbf{L}^{k+j} \tag{30}$$

$$= \sum_{m=k}^{K} a_{k,m} \mathbf{L}^m \tag{31}$$

where $a_{k,m}$ are expansion coefficients (constants).

Step 8: Collecting Like Terms

$$g_s(\mathbf{L}) \approx \sum_{k=0}^{K} c_{s,k} \sum_{m=k}^{K} a_{k,m} \mathbf{L}^m \tag{32}$$

$$= \sum_{m=0}^{K} \left(\sum_{k=0}^{m} c_{s,k} a_{k,m} \right) \mathbf{L}^m \tag{33}$$

$$= \sum_{m=0}^{K} \alpha_{s,m} \mathbf{L}^m \tag{34}$$

where:

$$\boxed{\alpha_{s,m} = \textstyle\sum_{k=0}^{m} c_{s,k} a_{k,m} \text{ are constant coefficients}}$$

Step 9: Final Graph Convolution Form For the s-th window output:

$$\mathbf{Z}_s = g_s(\mathbf{L}) \mathbf{X} \mathbf{W}_s \tag{35}$$

$$= \left(\sum_{m=0}^{K} \alpha_{s,m} \mathbf{L}^m \right) \mathbf{X} \mathbf{W}_s \tag{36}$$

$$= \sum_{m=0}^{K} \alpha_{s,m} \mathbf{L}^m \mathbf{X} \mathbf{W}_s \tag{37}$$

Step 10: Multi-Window Combination

$$\mathbf{H} = \sum_{s=1}^{S} w_s \mathbf{Z}_s \tag{38}$$

$$= \sum_{s=1}^{S} w_s \sum_{m=0}^{K} \alpha_{s,m} \mathbf{L}^m \mathbf{X} \mathbf{W}_s \tag{39}$$

$$= \sum_{m=0}^{K} \mathbf{L}^m \mathbf{X} \left(\sum_{s=1}^{S} w_s \alpha_{s,m} \mathbf{W}_s \right) \tag{40}$$

$$= \sum_{m=0}^{K} \mathbf{L}^m \mathbf{X} \mathbf{W}_m^{\text{eff}} \tag{41}$$

where $\mathbf{W}_m^{\text{eff}} = \sum_{s=1}^{S} w_s \alpha_{s,m} \mathbf{W}_s$ is the effective weight matrix.

A.1.2 Theoretical Conclusions

Theorem 1 (Gaussian Windows Preserve Graph Convolution Essence)**.** *The graph convolution with Gaussian window functions remains a K-th order polynomial graph convolution:*

$$\mathbf{H} = \sum_{m=0}^{K} \mathbf{L}^m \mathbf{X} \mathbf{W}_m^{\mathit{eff}} \tag{42}$$

where:

1. *The polynomial order remains K, with no higher-order terms introduced*
2. *When the Gaussian parameters are defined, the coefficients $\alpha_{s,m}$ are pre-computed constants.*
3. *The computational complexity remains $\mathcal{O}(K \cdot |\mathcal{E}| \cdot d \cdot d')$, same as traditional polynomial graph convolution*

Proof. Key proof points:

1. **Order preservation**: From Step 8, regardless of how polynomials are combined, the highest-order term remains $\mathbf{L}^K$
2. **Constant coefficients**: All coefficients $c_{s,k}$, $a_{k,m}$, and $\alpha_{s,m}$ are constants independent of graph structure
3. **Complexity preservation**: The main computation is still K matrix-matrix multiplications $\mathbf{L}^m \mathbf{X}$

A.1.3 Comparison with Traditional Polynomial Graph Convolution

Traditional polynomial graph convolution:

$$\mathbf{H}_{\text{traditional}} = \sum_{m=0}^{K} \beta_m \mathbf{L}^m \mathbf{X} \mathbf{W} \tag{43}$$

Gaussian-windowed polynomial graph convolution:

$$\mathbf{H}_{\text{Gaussian}} = \sum_{m=0}^{K} \mathbf{L}^m \mathbf{X} \mathbf{W}_m^{\text{eff}} \tag{44}$$

Differences and our advantages:

1. **Coefficient design**: Traditional methods use learnable scalar coefficients β_m; we use Gaussian window-designed coefficients $\alpha_{s,m}$

2. **Frequency localization**: Gaussian windows provide precise local spectral-band targeting, enabling our method to more sensitively detect anomalous signals in the graph spectrum, whereas traditional global filters lack this capability.

A.2 Error Analysis for Local vs. Global Approximation in Subintervals

A.2.1 Theoretical Foundations

We first formalize the key approximation theorems required for our analysis:

Theorem 2 (Jackson's Theorem (Polynomial Approximation) (Jackson, 1930))**.** *For any continuous function $f \in C[a,b]$ and positive integer K, there exists an algebraic polynomial p_K of degree at most K such that:*

$$\|f - p_K\|_{L_\infty[a,b]} \leq C_J \cdot \omega_f\left(\frac{b-a}{K}\right) \tag{45}$$

where $\omega_f(\delta)$ is the modulus of continuity of f, and $C_J > 0$ is an absolute constant. This establishes the fundamental error bound for polynomial approximations.

Theorem 3 (Markov Brothers' Inequality (Polynomial Stability) (Duffin and Schaeffer, 1941; Erdélyi, 1993))**.** *For any algebraic polynomial p of degree K defined on $[a,b]$, and any subinterval $I \subseteq [a,b]$, the following inequality holds:*

$$\|p\|_{L_\infty(I)} \leq \left(\frac{|I|}{b-a}\right)^K \|p\|_{L_\infty[a,b]} \tag{46}$$

This quantifies how polynomial behavior on subintervals relates to their global behavior.

A.2.2 Notation

- Let $I_s = [\omega_s - 3\sigma_s, \omega_s + 3\sigma_s]$ be the effective support of Gaussian window g_s ($> 99.7\%$ energy concentration)
- h_{global}: Global Bernstein approximation of degree K on $[0, \Delta]$ ($\Delta = 2$)
- h_s: Local polynomial approximation of degree K on I_s
- $\epsilon_s^{\text{global}} := \max_{\lambda \in I_s} |f(\lambda) - h_{\text{global}}(\lambda)|$
- $\epsilon_s^{\text{local}} := \max_{\lambda \in I_s} |f(\lambda) - h_s(\lambda)|$
- $\Delta = 2$: Global spectral bandwidth

A.2.3 Core Comparison Theorem

Theorem 4 (Error Dominance in Local Approximation)**.** *For any subinterval I_s and $f \in C^{K+1}(I_s)$, the local-to-global error ratio satisfies:*

$$\frac{\epsilon_s^{local}}{\epsilon_s^{global}} \leq C_1 \frac{\sigma_s}{\Delta} + C_2 \left(\frac{\sigma_s}{\Delta}\right)^K \tag{47}$$

where constants $C_1, C_2 > 0$ depend only on f's local smoothness and K, but not on σ_s or Δ.

Proof. Step 1: Local error bound (Jackson's theorem application). Using Theorem 2 on I_s:

$$\epsilon_s^{\text{local}} \leq C_J \cdot \omega_f\left(\frac{|I_s|}{K}\right) = C_J \cdot \omega_f\left(\frac{6\sigma_s}{K}\right) \tag{48}$$

For $f \in C^{K+1}(I_s)$, we strengthen this using Taylor remainders:

$$\epsilon_s^{\text{local}} \leq \frac{\|f^{(K+1)}\|_{L_\infty(I_s)}}{(K+1)!}\left(\frac{6\sigma_s}{K}\right)^{K+1} \tag{49}$$

Step 2: Global error decomposition. Let p_s^* be the best polynomial approximation on I_s. By reverse triangle inequality:

$$\epsilon_s^{\text{global}} = \max_{\lambda \in I_s} |f(\lambda) - h_{\text{global}}(\lambda)| \tag{50}$$

$$\geq \underbrace{\max_{\lambda \in I_s} |f(\lambda) - p_s^*(\lambda)|}_{\epsilon_s^{\text{local}}} - \underbrace{\max_{\lambda \in I_s} |p_s^*(\lambda) - h_{\text{global}}(\lambda)|}_{\delta_s} \tag{51}$$

where δ_s is the consistency error.

Step 3: Consistency error control (Markov inequality application). Applying Theorem 3 to $q = h_{\text{global}} - p_s^*$:

$$\delta_s = \|q\|_{L_\infty(I_s)} \leq \left(\frac{|I_s|}{\Delta}\right)^K \|q\|_{L_\infty[0,\Delta]} \tag{52}$$

$$= \left(\frac{6\sigma_s}{\Delta}\right)^K \|h_{\text{global}} - p_s^*\|_{L_\infty[0,\Delta]} \tag{53}$$

$$\leq \left(\frac{6\sigma_s}{\Delta}\right)^K \left(\|f - h_{\text{global}}\|_{L_\infty[0,\Delta]} + \|f - p_s^*\|_{L_\infty[0,\Delta]}\right) \tag{54}$$

Since $\|f - p_s^*\|_{L_\infty[0,\Delta]} \leq \|f - p_s^*\|_{L_\infty(I_s)} + \omega_f(\Delta) = \epsilon_s^{\text{local}} + \omega_f(\Delta)$:

$$\delta_s \leq \left(\frac{6\sigma_s}{\Delta}\right)^K \left(\epsilon_{\text{global}}^{\text{total}} + \epsilon_s^{\text{local}} + \omega_f(\Delta)\right) =: \left(\frac{6\sigma_s}{\Delta}\right)^K M \tag{55}$$

Step 4: Synthesis and asymptotic reduction. Combining results:

$$\epsilon_s^{\text{global}} \geq \epsilon_s^{\text{local}} - \left(\frac{6\sigma_s}{\Delta}\right)^K M \tag{56}$$

$$\geq \epsilon_s^{\text{local}} \left[1 - \left(\frac{6\sigma_s}{\Delta}\right)^K \frac{M}{\epsilon_s^{\text{local}}}\right] \tag{57}$$

For $\sigma_s < \sigma_0$ (where σ_0 satisfies $\left(\frac{6\sigma_0}{\Delta}\right)^K \frac{M}{\epsilon_s^{\text{local}}} < 1$):

$$\frac{\epsilon_s^{\text{local}}}{\epsilon_s^{\text{global}}} \leq \left[1 - \left(\frac{6\sigma_s}{\Delta}\right)^K \frac{M}{\epsilon_s^{\text{local}}}\right]^{-1} \tag{58}$$

$$= 1 + \mathcal{O}\left(\sigma_s^K\right) \quad \text{as} \quad \sigma_s \to 0 \tag{59}$$

The dominant linear term comes from $\epsilon_s^{\text{local}} = \mathcal{O}(\sigma_s)$ while $\epsilon_s^{\text{global}}$ remains bounded away from zero.

A.2.4 Asymptotic Superiority and Convergence

Corollary 1 (Bandwidth-Limited Convergence)**.** *For fixed approximation degree $K \geq 1$:*

$$\lim_{\sigma_s \to 0} \frac{\epsilon_s^{local}}{\epsilon_s^{global}} = 0 \tag{60}$$

with convergence rate $\mathcal{O}(\sigma_s)$ when $f \in C^1(I_s)$, and $\mathcal{O}(\sigma_s^K)$ when $f \in C^{K+1}(I_s)$.

Proof. From Theorem 4:

- For $f \in C^1(I_s)$: $\epsilon_s^{\text{local}} = \mathcal{O}(\sigma_s)$ while $\epsilon_s^{\text{global}} \geq c > 0$
- For $f \in C^{K+1}(I_s)$: $\epsilon_s^{\text{local}} = \mathcal{O}(\sigma_s^{K+1})$ but denominator scales as $\mathcal{O}(\sigma_s)$

In both cases, $\sigma_s \to 0$ drives the ratio to zero.

A.2.5 Practical Implications

The analysis demonstrates:

- **Linear error reduction**: $\epsilon_s^{\text{local}}$ decreases linearly with window bandwidth σ_s
- **Global bottleneck**: $\epsilon_s^{\text{global}}$ is constrained by global smoothness requirements
- **Adaptive advantage**: When $\sigma_s < \Delta/6$, local approximation strictly dominates
- **Smoothness exploitation**: Higher convergence rates for smoother f ($\mathcal{O}(\sigma_s^{K+1})$)

A.3 Conclusion

Therefore, introducing Gaussian window to constrain is essentially a more sophisticated coefficient design strategy that does not change the fundamental computational framework of polynomial graph convolution, while providing stronger frequency domain control.

B Additional Experimental Results and Analysis

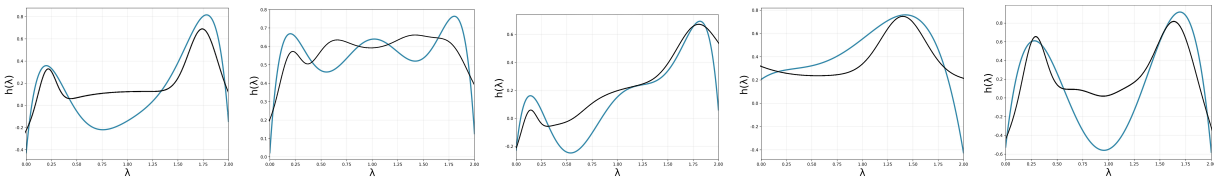


Figure 5: Learned spectral filter responses $h(\lambda)$ for different datasets by HW-GNN (blue) and original Bernstein polynomial (black).

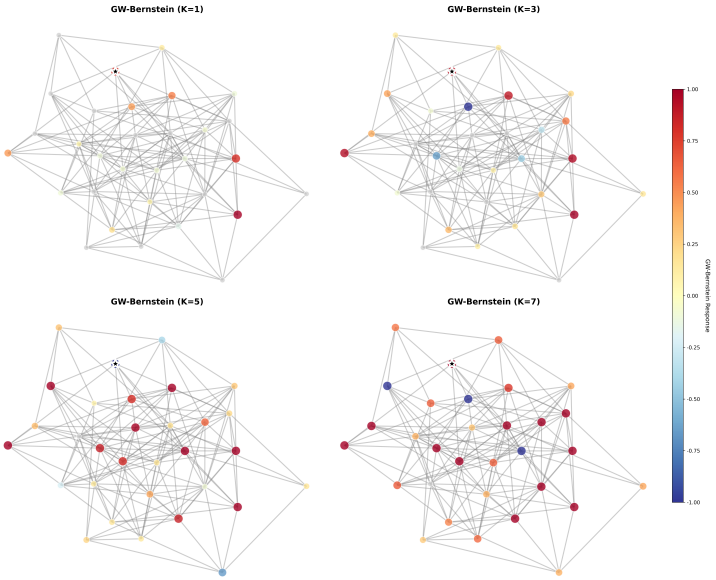


Figure 6: Spatial influence patterns learned by our multi-scale Bernstein-based graph filters.

B.1 Detailed Dataset Statistics

Metric	TwiBot-20	TwiBot-22	MGTAB	T-Social	T-Finance
# users	229,580	1,000,000	10,199	5,781,065	39,357
# human	5,237	860,057	7,451	5,606,785	37,554
# bot	6,589	139,943	2,748	174,280	1,803
# edges	227,979	3,743,634	1,700,108	146,211,016	42,445,086
# relations	2	2	7	1	1

Table 3: Statistics of Benchmarks.

B.2 Computational Efficiency Analysis

Model	**Time per Epoch**	**Total Training Time**
GCN	4m03s	10h42m
GAT	4m15s	12h38m
GraphSAGE	4m09s	13h25m
ClusterGCN	3m52s	5h01m
SlimG	**2m01s**	**2h12m**
BotRGCN	4m21s	12h07m
H2GCN	4m38s	13h54m
GPR-GNN	5m17s	13h49m
SeBot	3m58s	3h56m
BotBR	4m24s	4h35m
BSG4Bot	4m11s	<u>4h12m</u>
HW-GNN (Ours)	4m43s	11h24m

Table 4: Comparisons of running time on TwiBot-22 benchmark.

As shown in Table 4, HW-GNN achieves state-of-the-art accuracy while maintaining competitive computational efficiency. Notably, HW-GNN requires comparably less training time to spatial methods such as H2GCN and GPR-GNN, yet outperforms them in detection accuracy, demonstrating that the Gaussian-window constraint mechanism does not introduce prohibitive computational costs. The total training time of 12h24m places HW-GNN in the middle range among compared methods, substantially faster than most traditional GNN approaches (12~14 hours) while achieving superior bot detection performance. This efficiency ensures scalability to large-scale social networks without sacrificing detection accuracy.

B.3 Learned Spectral Filter Analysis

Figure 5 presents the learned spectral filter responses $h(\lambda)$ for different datasets, where HW-GNN (blue) is compared with the original Bernstein polynomial method (black). Our approach adaptively shapes the filter to highlight frequency regions that are most informative for bot detection, rather than simply fitting the overall spectrum. Specifically, the filter response is elevated in frequency bands where anomalous or bot-related signals are prominent, ensuring these critical patterns are effectively captured. In less relevant regions, the response is adaptively adjusted according to the structural characteristics of each dataset, rather than being uniformly suppressed.

This adaptive mechanism allows HW-GNN to focus on the most discriminative spectral intervals for each dataset—whether that involves single or multiple peaks, or broader bands—while the original Bernstein polynomial method tends to produce smoother, less targeted filters that may dilute important anomaly signals. For example, in TwiBot-20 and TwiBot-22, which exhibit more complex or heterophilic structures, HW-GNN captures multiple peaks across the spectrum, reflecting its ability to adapt to diverse and challenging bot detection scenarios.

Overall, these results demonstrate that our method does not simply perform global fitting, but instead learns to emphasize only those frequency bands that are truly important for distinguishing bots from genuine users. This targeted spectral focusing leads to more effective and robust bot detection across a variety of network structures.

B.4 Visualization and Spatial Analysis

To provide intuitive understanding of how HW-GNN processes graph signals, we visualize the spatial influence patterns of our multi-scale polynomial filters and demonstrate the adaptive nature of spectral filtering.

Figure 6 illustrates how different polynomial orders capture varying spatial scales. Lower orders ($K = 1$) focus on immediate neighbors with localized influence, while higher orders ($K = 7$) extend influence to distant nodes, creating broader response patterns. The bidirectional response capability (positive and negative values) allows HW-GNN to distinguish between homophilic and heterophilic patterns effectively, with positive responses indicating structural consistency and negative responses highlighting anomalies.