PA-GNN: Parameter-Adaptive Graph Neural Networks

Yuxin Yang[1] **Yitao Liang**[2 1] **Muhan Zhang**[2 1]

Abstract

Many influential areas require effective extraction and processing of graph information. Graph neural networks (GNNs) have been a type of powerful tools to obtain informative representations regarding both topology and node features. With an increasing number of graph properties being proposed and analyzed (such as homophily/heterophily, edge density, motifs, and feature distribution), numerous specific GNNs have been designed to capture them individually. However, most existing GNNs assume the entire graph shares the same property, and enforce parameter sharing across all regions of the graph. In this work, we introduce a novel class of GNNs which adopt a node-specific aggregation scheme with adaptive parameters. The node-specific parameters are generated according to node's neighborhood pattern and global position. By testing our model on semi-supervised node classification tasks on synthetic graphs and real-world benchmarks, we show its superiority over fixed-parameter models. The underlying idea could be applied as a flexible extension to different GNNs and solve a wide range of graph tasks.

1. Introduction

Graph structured data have been widely seen in many influential areas, e.g. bioinformatics (Jiang et al., 2021; Rathi et al., 2019), social network analysis (Qiu et al., 2018; Li & Goldwasser, 2019), and recommendation systems (Ying et al., 2018; Wang et al., 2019). As an irregular non-Euclidean data structure, graph typically contains information of both node attributes and graph structure. Graph data mining techniques have been utilized to extract knowledge from graphs to fulfill various downstream tasks, such as semi-supervised node classification (Kipf & Welling, 2017), link prediction (Zhang & Chen, 2018), and graph classification (Xu et al., 2019). Recently, graph neural networks have emerged as an effective approach for these tasks.

The objective of employing GNNs is to obtain an integrated representation of graph topology, node attributes, and sometimes edge attributes. The representation generation usually involves a few iterations. In each iteration nodes aggregate information from their ego-embeddings and their neighbors' embeddings. The aggregation scheme is a major difference among various GNNs. Most existing models adopt aggregation schemes equivalent to low-pass filters. They essentially perform a local Laplacian smoothing (Xu et al., 2019; Kipf & Welling, 2017; Zhu et al., 2020). Zhu et al. (2020) defines a measurement of edge homophily. By revealing that homophily is not a trivial assumption, this work has inspired researchers to find more universal GNN architectures which can effectively aggregate information in both homophilic and heterophilic graphs. From a spectral perspective, GNNs have been extended from rigid low-pass filters to learnable arbitrary filters (Chien et al., 2020; Luan et al., 2021; He et al., 2021; Wang & Zhang, 2022).

In order to test GNN performance on graphs with different properties, a variety of graph benchmarks have been developed. Recent works focus on different label (Zhu et al., 2020) and feature distributions (Palowitch et al., 2022) as well as enlarged graph scale (Hu et al., 2020). However, we focus on an otherwise overlooked phenomenon: the disparity of regional patterns in graphs. That is, different regions of a graph might have much different properties. To make use of this characteristic, we take on the idea of dynamic neural networks. Dynamic neural networks are able to adjust their architecture or parameters according to the input, leading to higher efficiency, stronger representation power and interpretability (Han et al., 2021).

In this work, we propose to learn node-specific parameters in order to adapt to regional patterns. Based on GPRGNN (Chien et al., 2020), we propose a parameter-adaptive graph neural network (PA-GNN), which augments ordinary GNNs with an auxiliary parameter-generating network. We demonstrate the effectiveness of PA-GNN on both synthetic graph datasets and semi-supervised node classification benchmarks. Our PA-GNN has comparable results on homophily graphs, and exceeds the baseline GPRGNN on

[1]Beijing Institute for General Artificial Intelligence [2]Institute for Artificial Intelligence, Peking University. Correspondence to: Muhan Zhang <muhan@pku.edu.cn>.

DynNN workshop at the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

all heterophily benchmark datasets. Finally, we introduce a real-world graph dataset where our model remarkably outperforms baseline models.

2. Related Works

Graph neural networks as low-pass filters. Many spectral-designed (Defferrard et al., 2016; Levie et al., 2018) graph neural networks ground themselves on spectral graph theory. Some are designed as low-pass filters whose spectral response stays invariant across graphs. Some other GNNs are designed spatially, but they have spectral interpretations too. For example, GCN (Kipf & Welling, 2017), APPNP (Klicpera et al., 2019), GIN (Xu et al., 2019) and GAT (Velickovic et al., 2018) behave like low-pass filters according to the analysis of Balcilar et al. (2021).

Beyond graph homophily assumptions. Homophily or heterophily refer to the tendency for nodes from the same class to connect or disconnect. GNNs which are low-pass filters perform well on homophilic graphs due to the smoothing effect, but they make class-specific embeddings indiscriminative on heterophilic graphs. Recently, revisiting the homophily assumption aroused interest in developing GNNs with spectral adaptibility. Zhu et al. (2020) defined edge homophily H_{edge} to measure graph homophily level. H_{edge} is defined as the proportion of inter-class edges over all edges. Follow-up works invent other criteria to measure graph homophily level, including node homophily ratio H_{node} (Pei et al., 2020) and class homophily H_{class} (Lim et al., 2021). These works state that high and low homophily levels require low-pass and high-pass filters respectively. Many graph neural networks are hence designed to learn adaptive passband. ACMII (Luan et al., 2021) assigns channels to low- and high-frequency signals separately. ChebyNet (Defferrard et al., 2016) proposes to use the ChebyShev polynomial to approximate arbitrary filters. GPRGNN (Chien et al., 2020) directly learns individual parameters as the polynomial coefficients. BernNet (He et al., 2021) uses Bernstein polynomials and also learn the coefficients. Recently, Wang & Zhang (2022) unify existing spectral GNNs and analyze their expressive power. However, existing works assume all regions of the graph share the same spectral property and applies the same filters to all nodes.

Dynamic neural networks. The concept of dynamic neural networks (DNNs) has been systematically explored in (Han et al., 2021). Dynamic parameters networks are one category of DNNs that lift the performance efficiently. Some DNNs generate parameters directly from input, while others utilize task-specific information as well. For example, Kang et al. (2017) leverage camera perspective to generate kernel weights for image convolution. With regard to graphs, a few recent works show the effectiveness of GNNs with adaptive structure or parameters. Wu et al. (2019) adopt degree-specific weights and hashing functions assisted by special graph-level readout functions to learn graph representations in a Hilbert kernel space. Simonovsky & Komodakis (2017) utilize edge attributes to adapt weights for performing convolution on graphs.

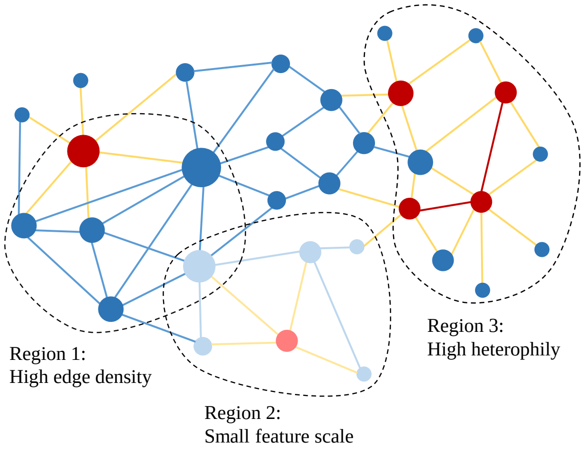


Figure 1: Graph with different regional patterns. Node size increases with degree. Two classes are represented by red (light red) and blue (light blue). Deeper node color represents larger feature scale. Blue and red edges connect same-class nodes, while yellow edges connect different-class nodes. We observe regions with high node density, small node feature scale and high heterophily in the graph.

3. Parameter-Adaptive Graph Neural Networks

In this section, we first motivate learning node-adaptive parameters. Then, we introduce our parameter-adaptive graph neural network (PA-GNN) framework, including the network structure, input and output. Finally, we explain how our model is able to cope with complicated situations.

3.1. Advantages of dynamic parameter generating

In Figure 1, the graph has different subgraph patterns; the three circled regions have regional patterns that is different from the center part of the graph: In region 1 with high edge density, the average node degree is above 5, which might indicate more complex structural patterns to learn and higher noise levels. In region 2, the significantly smaller feature scale might indicate the need to use different parameter scales. In region 3 with high heterophily, an aggregator as a high-pass filter will perform much better than a low-pass one. For this graph, a uniform aggregation scheme is suboptimal for learning node representations for node classification, bringing adaptive aggregation into our sight. Intuitively, global positional information can be used to recognize the region to which a node belongs, while local structures contain region-specific graph properties.

3.2. Notations

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph. $\mathcal{V}$ and $\mathcal{E}$ are the node and edge sets of the graph, and the node number $n = |V|$. $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix, $\tilde{A} = A + I$ represents the adjacency matrix with self loops. $\tilde{D}$ is the degree matrix of $\tilde{A}$. $\tilde{A}_{sym} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ stands for the normalized adjacency matrix with self loops. $X \in \mathbb{R}^{n \times d}$ is the initial node feature matrix.

3.3. Methodology

Our model consists of two major components: a parameter network and a classification network. Our primary contribution is that we use the parameter network to generate node-specific aggregation parameters for the classification network. The pipeline is three-step (see Figure 2): first, the graph structure is used to produce latent positional embeddings to augment node features; second, the parameter network generates node-specific parameters based on the augmented input; third, the classification network takes the generated parameters to predict node classes. We implement both networks based on GPRGNN.

Adaptive universal generalized PageRank. The building blocks of our model is GPRGNN. The Generalized PageRank GNN (GPRGNN) model is proposed by Chien et al. (2020). Given the initial node embedding $H^{(0)} \in \mathbb{R}^{n \times d}$, the output node embedding is calculated by

$$Z = \sum_{k=0}^{K} \gamma_k \tilde{A}_{sym}^k H^{(0)}. \tag{1}$$

The added self-loops in the adjacency matrix alleviate the over-smoothing problems of GNNs. The work proves that different GPR weights are related to the network's passbands from the spectral perspective. With learnable γ_k, GPRGNN is able to simulate arbitrary passbands in the message passing process (with large enough K), thus leading to good performance on heterophilic graphs.

Classification network. The classification network is implemented as an N-layer GPRGNN. This network yields the embedding for classification from two inputs: the initial node embedding $H^{(0)} = X \in \mathbb{R}^{n \times d}$ and the weights $\gamma_{ik} \in \mathbb{R}, i \in \mathcal{V}, k = 0, 1, 2, ..., N$. The output embedding of node i for classification is given by $Z_i = \sum_{k=0}^{N} \gamma_{ik} [\tilde{A}_{sym}^k H^{(0)}]_i$.

Parameter Network. The parameter network is used to generate the node-specific GPR weights γ_{ik}. To prevent overfitting, we decompose γ_{ik} into $\gamma_{ik} = \gamma_k + \Delta\gamma_{ik}$, where γ_k is learnable parameter shared by all nodes and $\Delta\gamma_{ik}$ is the node-specific weight offset output by the parameter network. We use another M-layer GPRGNN to implement the parameter network. Let $H_p^{(0)} \in \mathbb{R}^{n \times d}$ denote the node features fed into the parameter network. The output embedding is calculated as $Z_p = \sum_{k=0}^{M} \varphi_k \tilde{A}_{sym}^k H_p^{(0)}$. Then, node i's embedding $[Z_p]_i$ is passed through an MLP to produce $\Delta\gamma_{ik}, k = 0, 1, 2, ..., N$.

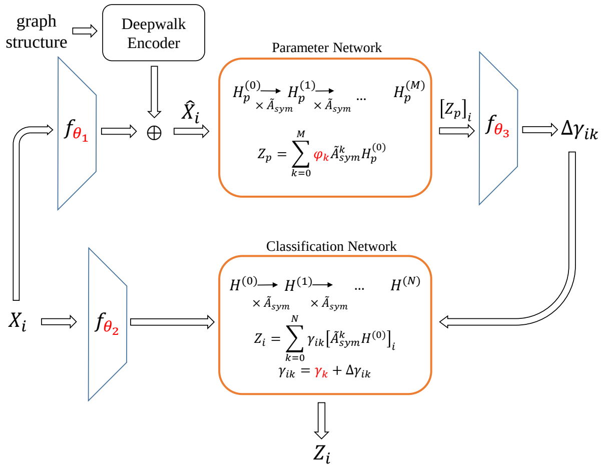

graph structure
Deepwalk Encoder
Parameter Network
$H_p^{(0)} \xrightarrow{\times \tilde{A}_{sym}} H_p^{(1)} \xrightarrow{\times \tilde{A}_{sym}} \dots \quad H_p^{(M)}$
$Z_p = \sum_{k=0}^{M} \varphi_k \tilde{A}_{sym}^k H_p^{(0)}$
f_{θ_1}
$\oplus$
$\hat{X}_i$
$[Z_p]_i$
f_{θ_3}
$\Delta\gamma_{ik}$
Classification Network
$H^{(0)} \xrightarrow{\times \tilde{A}_{sym}} H^{(1)} \xrightarrow{\times \tilde{A}_{sym}} \dots \quad H^{(N)}$
$Z_i = \sum_{k=0}^{N} \gamma_{ik} [\tilde{A}_{sym}^k H^{(0)}]_i$
$\gamma_{ik} = \gamma_k + \Delta\gamma_{ik}$
X_i
f_{θ_2}
Z_i

Figure 2: Pipeline of our proposed model PA-GNN. Blue trapezoids represent stacked linear layers. Red colored symbols denote learnable parameters. Node feature X is concatenated with DeepWalk embedding to produce input $H_p^{(0)}$ to the parameter network. $H^{(k)}$ denotes hidden states by GPR propagation. The parameter network produces node-specific parameters $\Delta\gamma_{ik}$, where i is node index. These parameters are added to a learnable node-invariant weight γ_k to produce the GPR weights of the classification network. The eventual representations of node i, Z_i, is the weighted sum of hidden states $[H^{(k)}]_i, k = 0, 1, 2, ..., N$.

Encoding global and local information The input $H_p^{(0)}$ to the parameter network is the initial node feature vector augmented by positional node embeddings, which are generated with DeepWalk (Perozzi et al., 2014). In DeepWalk, nodes that connect to each other or have short paths between them will be close in the latent embedding space, which encodes the region information to which a node belongs. $H_p^{(0)}$ consists of two concatenated parts: initial node feature vector goes through an MLP to reduce its dimension, and a DeepWalk encoder extracts latent positional information from the graph structure.

Joint training of both networks The two networks are simultaneously optimized through gradient descent. Besides the classification loss, we regularize the network with the L_1-norm of γ_{ik}. Larger regularization prevents node-specific weights γ_{ik} from varying too much among different nodes, and helps PA-GNN learn better on simple graphs by enforcing $\gamma_{ik} \approx \gamma_k$. On complicated graphs, we may loosen the regularization to improve the adaptability of our model.

Table 1: Results on benchmark datasets: mean test accuracy (%) ± 95% confidence interval. The best results are in bold. Results within the confidence interval are underlined.

	Cora	Citeseer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Twitch
GCN	75.21±0.38	67.30±0.35	84.27±0.01	82.52±0.32	90.54±0.21	60.96±0.78	30.59±0.23	45.66±0.39	45.68±0.46
GAT	76.70±0.42	67.20±0.46	83.28±0.12	81.95±0.38	90.09±0.27	63.90±0.46	35.98±0.23	42.72±0.33	46.32±0.07
GCN-Cheby	71.39±0.51	65.67±0.38	83.83±0.12	82.41±0.28	90.09±0.28	59.96±0.51	38.02±0.23	40.67±0.31	45.51±0.45
JKNet	73.22±0.64	60.85±0.76	82.91±0.11	77.80±0.97	87.70±0.70	62.92±0.49	33.41±0.25	44.72±0.48	46.03±0.48
APPNP	79.41±0.38	**68.59±0.30**	85.02±0.09	81.99±0.26	91.11±0.26	51.91±0.56	38.86±0.24	34.77±0.34	39.98±1.24
GPRGNN	**79.51±0.36**	67.63±0.38	**85.07±0.09**	82.90±0.37	91.93±0.26	67.48±0.40	39.30±0.27	49.93±0.53	45.90±0.27
PA-GNN	79.42±0.35	66.89±0.36	85.04±0.08	**83.93±0.50**	**91.95±0.37**	**68.14±0.48**	**39.33±0.22**	**51.21±0.71**	**50.64±0.14**

4. Experiments

We perform semi-supervised node classification on both synthetic graphs and real-world benchmark datasets.

4.1. Evaluation on synthetic graphs

We use contextual stochastic block models (cSBMs) (Deshpande et al., 2018) to generate large graphs composed of subgraphs with different properties, in order to synthesize graphs with mixed regional patterns. On the large graphs, we compare our model with two baselines.

We use the original GPRGNN as the first baseline. We also propose a baseline which first divides the graph into different clusters using spectral clustering, then trains a fixed-parameter GPRGNN on each of the clusters. We name this baseline CLGNN. CLGNN is supposed to perform well when inter-subgraph edges are sparse. It is worth mentioning that by using this model, the labeled training and test samples will be in inverse proportion to the cluster number for each GPRGNN, which might lead to degraded performance. Due to space limit, the dataset and experiment details are covered in Appendix A. Our experiment reveals that our PA-GNN framework consistently exceeds fixed-parameter GNNs and cluster-specific GNNs in classification accuracy.

4.2. Evaluation on real-world graph datasets

Testing on benchmark datasets We test our framework on benchmark datasets. We use 5 homophilic datasets: Citation networks Cora, Citeseer, PubMed (Sen et al., 2008; Yang et al., 2016) and Amazon co-purchasing networks Computers and Photo (McAuley et al., 2015; Shchur et al., 2018). We also use 3 heterophilic datasets: Wiki page-page network Chameleon and Squirrel (Rozemberczki et al., 2021), and actor co-occurence network Actor. We summarize the graph characteristics in Table 2 in the appendix, and present the results in Table 1. Our model exceeds GPRGNN on most benchmarks, which is partly explained by that GPRGNN is a special case of PA-GNN when all nodes share the same classification network parameters.

We also introduce a new graph dataset Twitch by extracting a subset from Twitch Gamers (Rozemberczki & Sarkar, 2021), a game streaming social network dataset, in which nodes represent users and edges represent mutual following relationship. Node attributes contain information about account status, content and language. Twitch shows uneven regional properties when divided by language, thus is a perfect example to demonstrate the advantage of learning node-specific parameters. We statistically show the nation-wise behaviours of users in Appendix B, including different edge density and homophily levels. On Twitch, our model PA-GNN exceeds GPRGNN and other baselines remarkably. The outstanding performance may indicate that our model has captured more nation-specific information to aid the node classification task. Results of ablation study is shown in Appendix C.

We want to emphasize that Twitch is not just an exception, but a representative showing how diversified local graph patterns can be. As we expand our vision to larger-scale graphs, the regional disparity should be more visible and important. For example, the graph patterns may be significantly different between genders, among races and countries. Naively dividing by clusters may result in insufficient examples for each cluster, while our PA-GNN framework can jointly leverage all the examples and adapt to different regions.

5. Conclusions

We reveal an overlooked phenomenon of graph datasets: the disparity of regional patterns. Current GNNs are not designed to adapt to different local patterns by using a uniform aggregation scheme for all nodes. We design a model that can generate node-specific aggregation parameters from augmented node features. Our model shows superior performance on both synthetic and real-world datasets. To better demonstrate the effectiveness of our model, we introduce a new graph dataset Twitch, on which our model exceeds all other models with a large margin. While we exhibit a relatively simple architecture and only use DeepWalk to encode positional information, other properties may also be utilized to better generate the parameters, including motifs, density, etc. Furthermore, our framework can take other spectral- or spatial GNNs as well and can be extended to other tasks including link prediction and graph classification.

References

Balcilar, M., Renton, G., Héroux, P., Gaüzère, B., Adam, S., and Honeine, P. Analyzing the expressive power of graph neural networks in a spectral perspective. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Chien, E., Peng, J., Li, P., and Milenkovic, O. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations*, 2020.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3837–3845, 2016.

Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. Contextual stochastic block models. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8590–8602, 2018.

Han, Y., Huang, G., Song, S., Yang, L., Wang, H., and Wang, Y. Dynamic neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

He, M., Wei, Z., Xu, H., et al. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34, 2021.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Jiang, D., Wu, Z., Hsieh, C.-Y., Chen, G., Liao, B., Wang, Z., Shen, C., Cao, D., Wu, J., and Hou, T. Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13 (1):1–23, 2021.

Kang, D., Dhar, D., and Chan, A. B. Incorporating side information by adaptive convolution. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 3867–3877, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Klicpera, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

Li, C. and Goldwasser, D. Encoding social information with graph convolutional networks forpolitical perspective detection in news media. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2594–2604, 2019.

Lim, D., Li, X., Hohne, F., and Lim, S.-N. New benchmarks for learning on non-homophilous graphs. *ArXiv preprint*, abs/2104.01404, 2021.

Luan, S., Hua, C., Lu, Q., Zhu, J., Zhao, M., Zhang, S., Chang, X.-W., and Precup, D. Is heterophily a real nightmare for graph neural networks to do node classification? *ArXiv preprint*, abs/2109.05641, 2021.

McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.

Palowitch, J., Tsitsulin, A., Mayer, B., and Perozzi, B. Graphworld: Fake graphs bring real insights for gnns. *ArXiv preprint*, abs/2203.00112, 2022.

Pei, H., Wei, B., Chang, K. C., Lei, Y., and Yang, B. Geomgcn: Geometric graph convolutional networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., and Tang, J. Deepinf: Social influence prediction with deep learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2110–2119, 2018.

Rathi, P. C., Ludlow, R. F., and Verdonk, M. L. Practical high-quality electrostatic potential surfaces for drug discovery using a graph-convolutional deep neural network. *Journal of medicinal chemistry*, 63(16):8778–8790, 2019.

Rozemberczki, B. and Sarkar, R. Twitch gamers: a dataset for evaluating proximity preserving and structural role-based node embeddings. *arXiv preprint arXiv:2101.03091*, 2021.

Rozemberczki, B., Allen, C., and Sarkar, R. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Simonovsky, M. and Komodakis, N. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 29–38. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.11.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., and Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 968–977, 2019.

Wang, X. and Zhang, M. How powerful are spectral graph neural networks. *arXiv preprint arXiv:2205.11172*, 2022.

Wu, J., He, J., and Xu, J. Demo-net: Degree-specific graph neural networks for node and graph classification. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G. (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 406–415. ACM, 2019. doi: 10.1145/3292500.3330950.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

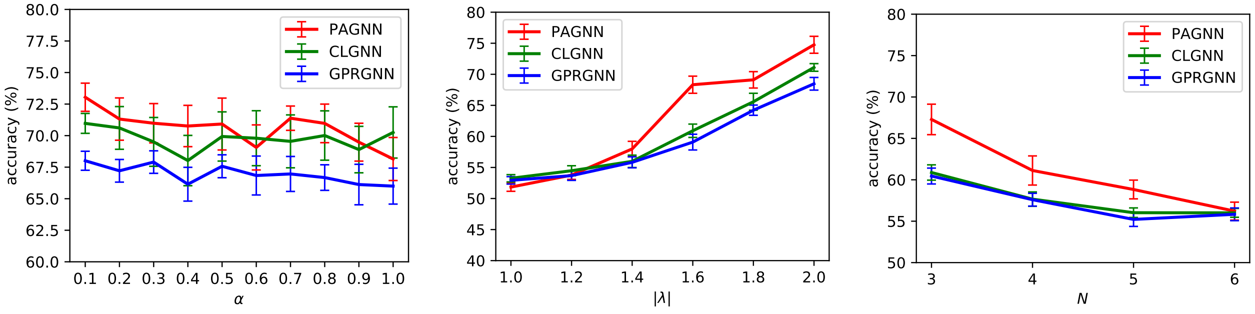
Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

A. Experiment on synthetic datasets.



(a) Control α.

(b) Control homophily h.

(c) Control subgraph number N.

Figure 3: Figures (a) to (c) show the accuracy of our model and two baseline models on synthetic datasets.

A.1. cSBM Synthetic Dataset

Contextual Stochastic Block Model (cSBM) is a synthetic graph generating algorithm. It models the graph as connected communities, each community is viewed as a different class. The node features are randomly sampled from class-assigned Gaussian distributions. A parameter μ controls the divergence of the Gaussian means. λ controls inter-class and intra-class densities. Positive and negative λ induce homophily and heterophily in graphs respectively. The synthetic graph has assignable average node degree d and feature dimensions p.

A.2. Experiment settings and results.

We generate subgraphs of different edge homophily level H_{edge} and combine them into a whole graph. Similar to the definition of inter-class and intra-class edges, we refer to the edges as inter-subgraph and intra-subgraph edges. We use a parameter α to represent the ratio of inter-subgraph edges over all edges. For all experiments, we use 2.5% data for training, 2.5% data for validation, and 95% data for test.

We display the performance of the three models regarding different α. There are two equal-size classes in each of the two subgraphs, with $\lambda = 2, -2$. As α increases from 0.1 to 1, inter-subgraph edges increase and subgraphs are gradually blended. PA-GNN consistently outperforms GPRGNN by a large margin (see Figure 3(a)). CLGNN achieves close accuracies with PA-GNN in general.

We mix two subgraphs and control the homophily level of them by varying λ. We set $\alpha = 0.5$, and set two subgraph λ's inverse numbers. The pattern of the merged graph is controlled by a single parameter $|\lambda|$. Bigger $|\lambda|$ means the graph has richer sturctural information. Our model PA-GNN exceed baseline methods when $|\lambda|$ is bigger (see Figure 3(b)), showing the model's strong ability to capture graph topology.

We also alter subgraph number N to see how our method generalize to different settings. We set $\alpha = 0.5, \lambda_{min} = 1.2, \lambda_{max} = 1.8$. The other λ values are equally spaced in between. Figure 3(c) shows that PA-GNN surpasses the baselines. With a large number of connecting subgraphs, the whole graph evolves into a mixed component, and local patterns become indistinguishable. This may count for the reducing performance gap as N increases.

B. Statistics of Real-World Datasets

	Cora	Citeseer	PubMed	Computers	Photo	Chameleon	Squirrel	Actor	Twitch
#nodes	2708	3327	19717	13752	7650	2277	5201	7600	43703
#edges	5278	4552	44324	245861	119081	31371	198353	26659	1325164
#features	1433	3703	500	767	745	2325	2089	932	7
#classes	7	6	5	10	8	5	5	5	3
H_{edge}	0.825	0.718	0.792	0.802	0.849	0.247	0.217	0.215	0.403

Table 2: Benchmark datasets properties and statistics

language	NO	FI	RU	ES	SV	TH	IT	JA	KO	PT
#nodes	330	652	4821	5699	854	632	1230	1327	1215	2536
H_{edge}	0.364	0.404	0.372	0.369	0.399	0.604	0.370	0.377	0.616	0.352
language	NL	TR	DA	ZH	CS	PL	HU	FR	DE	OTHER
#nodes	701	772	503	2828	576	944	427	6799	9428	1429
H_{edge}	0.358	0.323	0.362	0.556	0.364	0.411	0.402	0.385	0.367	0.401

Table 3: Homophily level using target life-time in ground-truth language communities.

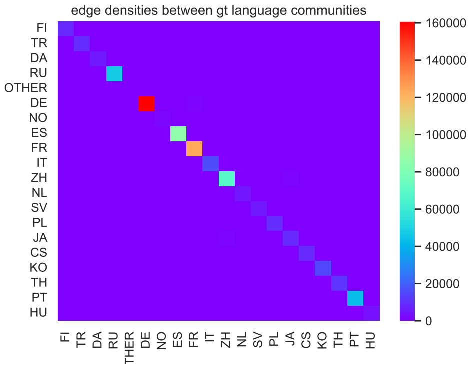


Figure 4: In Twitch Gamers dataset, we demonstrate the densities of links between different ground-truth communities grouped by language. The intra-class links outnumber inter-class links.

B.1. Existing benchmark datasets

We test our framework on some benchmark datasets. We use 5 homophilic datasets: Citation networks Cora, Citeseer, PubMed (Sen et al., 2008; Yang et al., 2016) and Amazon co-purchasing networks Computers and Photo (McAuley et al., 2015; Shchur et al., 2018). We also use 3 heterophilic datasets: Wiki page-page network Chameleon and Squirrel (Rozemberczki et al., 2021), and actor co-occurence network Actor. We summarize the graph characteristics in Table 2.

B.2. Twitch Gamers: a game streaming network graph dataset

Twitch Gamers is a global game streaming dataset (Rozemberczki & Sarkar, 2021). The nodes represent users and edges represent mutual following relationship. Node attributes contain information about account status, content and language. We remove all accounts with English broadcast language to reduce the size of graph. To produce the classification targets, We divide account life-time into 3 equally-sized categories. We use the other features to predict. Statistics are shown in Table 2.

The Twitch dataset graph can be well-divided with the language attribute. As shown in Figure 4, we observe that the mutual following relationship often happens between same language users, and different languages also show significantly different edge densities. We further analyze the edge homophily level H_{edge} in each community seperately, and found a diversified distribution (shown in Table 3). These statistics demonstrates the regional disparity of homophily level on this graph.

C. Ablation study

node features	DeepWalk embeddings	Cora	Citeseer	PubMed	Computers	Photo	Chameleon	Actor	Squirrel	Twitch
✓	✓	79.42±0.35	66.89±0.36	85.04±0.08	83.93±0.50	91.95±0.37	68.14±0.48	39.33±0.22	51.21±0.71	50.64±0.14
✓	-	79.48±0.38	66.85±0.53	84.57±0.89	84.09±0.34	91.98±0.24	67.49±0.40	39.59±0.25	52.15±0.58	46.64±0.08
-	✓	79.20±0.33	67.15±0.38	84.16±1.25	83.58±0.52	92.17±0.18	67.95±0.54	39.31±0.22	52.87±0.34	50.62±0.16

Table 4: Ablation study results (mean test accuracy (%) ± 95% confidence interval) on PA-GNN on benchmark datasets with different input to the parameter network.

We conduct ablation study on the input $H_p^{(0)}$ to the parameter network. We compare classification accuracy with different input (Table 4) to the parameter network. We draw two conclusions from the results: First, some graphs show different local patterns while other do not. Second, on the graphs with regional disparity, node position information mainly contributes to generating optimal parameters for PA-GNN.