Second-Order Global Attention Networks for Graph Classification and Regression

Fenyu Hu[1,2](✉), Zeyu Cui[3], Shu Wu[1,2], Qiang Liu[1,2], Jinlin Wu[1,2], Liang Wang[1,2], and Tieniu Tan[1,2]

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
fenyu.hu@cripac.ia.ac.cn,
{shu.wu,qiang.liu,jinlin.wu,wangliang,tnt}@nlpr.ia.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] DAMO Academy, Alibaba Group, Hangzhou, China

Abstract. Graph Neural Networks (GNNs) are powerful to learn representation of graph-structured data, which fuse both attributive and topological information. Prior researches have investigated the expressive power of GNNs by comparing it with Weisfeiler-Lehman algorithm. In spite of having achieved promising performance for the isomorphism test, existing methods assume overly restrictive requirement, which might hinder the performance on other graph-level tasks, e.g., graph classification and graph regression. In this paper, we argue the rationality of adaptively emphasizing important information. We propose a novel global attention module from two levels: channel level and node level. Specifically, we exploit second-order channel correlation to extract more discriminative representations. We validate the effectiveness of the proposed approach through extensive experiments on eight benchmark datasets. The proposed method performs better than the other state-of-the-art methods in graph classification and graph regression tasks. Notably, It achieves 2.7% improvement on DD dataset for graph classification and 7.1% absolute improvement on ZINC dataset for graph regression.

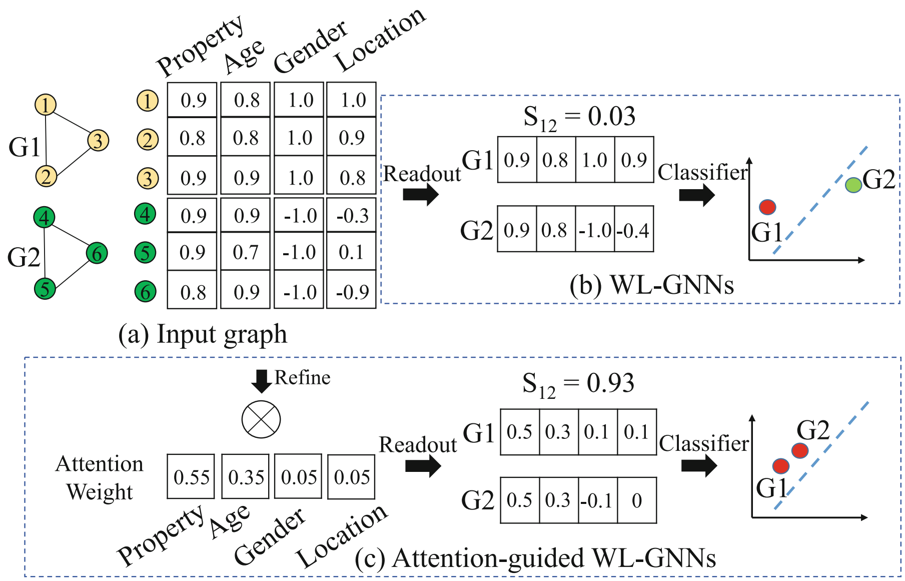
Keywords: Graph classification · Graph regression · Graph neural networks · Attention mechanism

1 Introduction

Graph Neural Networks (GNNs) have proved to be powerful in learning representation of graph data and have attracted a surge of interests [1,3,7,8,25,28–32]. Recently, numerous approaches have been proposed to quantify such representation power of GNNs [18,20,27]. These approaches try to bridge a theoretical connection with the Weisfeiler-Lehman (WL) algorithm [24] when judging the graph isomorphism. We term these approaches as WL-GNNs. In general, WL-GNNs

S. Wu—To whom correspondence should be addressed.

L. Fang et al. (Eds.): CICAI 2022, LNAI 13605, pp. 496–507, 2022.
https://doi.org/10.1007/978-3-031-20500-2_41



	Property	Age	Gender	Location
1	0.9	0.8	1.0	1.0
2	0.8	0.8	1.0	0.9
3	0.9	0.9	1.0	0.8
4	0.9	0.9	-1.0	-0.3
5	0.9	0.7	-1.0	0.1
6	0.8	0.9	-1.0	-0.9

G1	0.9	0.8	1.0	0.9
G2	0.9	0.8	-1.0	-0.4

Property	Age	Gender	Location
0.55	0.35	0.05	0.05

G1	0.5	0.3	0.1	0.1
G2	0.5	0.3	-0.1	0

Fig. 1. A toy example of predicting the debt-paying ability of different companies. $G1$ and $G2$ show the shareholder relationships of two companies, which should be classified as the same class. (a) Each node represents a shareholder of the company and consists of four numerical attributes. These two graphs are quite similar in the first two important attributes but have vast difference in the latter two trivial attributes. (b) $G1$ and $G2$ are distinguished by WL-GNNs and may be misclassified into different classes. S_{12} refers the cosine similarity of $G1$ and $G2$. (c) Attention-guided WL-GNNs yield much closer distance in latent space.

aim to distinguish non-isomorphic graphs by approximating an injective hash function. More specifically, if two graphs are non-isomorphic, they are expected to have different embeddings through WL-GNNs. Owing to the superior performance of WL-GNNs in distinguishing graphs with regard to isomorphism test, there are also approaches that apply WL-GNNs in graph classification and graph regression tasks [18,20,21,27].

Although WL-GNNs have shown strong advantages in above graph analytical tasks, a deficiency is that graph isomorphism property is an overly restrictive requirement for other graph-level tasks. Specifically, *since any difference between two graphs can lead to the non-isomorphism, WL-GNNs do not need to consider feature importance*. However, this makes WL-GNNs inadequate to discover determinative signals that indicate the graph characteristics in some graph-level tasks. Figure 1 illustrates how WL-GNNs limit the performance in these tasks, where we consider a toy problem of predicting the debt-paying ability of different companies. Suppose $G1$ and $G2$ are two small companies, where nodes denote shareholders and the edges denote partnership. Given four dimensions representing the *property*, *age*, *gender* and *location* of the shareholders, it is commonly accepted that the rich and the middle-aged will have a higher chance to repay a loan than the poor and the young. Therefore, personal property and age features should contribute relatively more than the other two features in estimating the solvency. Correspondingly, in Fig. 1(a), $G1$ and $G2$ with similar property and age features are expected to behave similarly. Nevertheless, since $G1$ and $G2$ have obvious difference in gender and location features, the cosine similarity between these two companies is only 0.03. As shown in Fig. 1(b), when applying conventional WL-GNNs with injective multiset function [27], $G1$ and $G2$ may be far away from each other in the latent representation space and be misclassified into different classes.

A simple way to infer the global channel attention weights of graphs is employing an average pooling aggregator at each channel, and then apply some learning mechanisms to obtain the attention weights (illustrated in Fig. 1(c)). However, the global average pooling operation only explores first-order statistics, ignoring channel interdependencies. Taking Fig. 1 as an example, there might be strong interdependencies between personal property and age, because one usually needs decades of years to accumulate its fortune. Hence, we need adequately considering graph channel interdependencies. Recent works in computer vision have also shown that deep neural networks with such second-order statistics can improve classification performance [16,17,26]. To this end, we are inspired to develop a novel second-order global channel attention network to fully exploit the channel interdependencies. Similarly, global node attention also helps extract important information (please refer to Sect. 3.2 for more details).

On this basis, we propose a *Second-order Global Channel Attention* (SoGCA) mechanism for better channel correlation and importance learning. Our SoGCA adaptively learns important information by exploiting second-order channel statistics, extracting more discriminative representations. Moreover, a *Global Attention-guided Structure* (GAS) is presented to highlight important information from two levels: channel level and node level. By stacking GAS after each graph isomorphism aggregator, we obtain a *Second-order Global Attention Network* (SGAN) which is compatible with existing WL-GNNs. In order to evaluate the generality of SGAN, we devise three variants based on GIN [27], 3WLGNN [18] and PNA [2], respectively. We conduct comprehensive experiments on eight public datasets and achieve state-of-the-art results on all benchmark tasks.

2 Preliminaries

2.1 Notations and Problem Definition

Consider a graph $G(\mathcal{V}, \mathcal{E}, \mathbf{X})$ with $N = |\mathcal{V}|$ nodes and $|\mathcal{E}|$ edges. $\mathbf{X} \in \mathbb{R}^{N \times d_0}$ denotes the node feature matrix, where d_0 is the number of input attributes. Given a collection of graphs $\{G_1, ..., G_N\}$ and their corresponding labels $\{y_1, ..., y_N\}$, the task of graph classification or graph regression is to learn a mapping $f : \mathcal{G} \rightarrow \mathcal{Y}$, where $\mathcal{G}$ is the set of input graphs and $\mathcal{Y}$ is the set of labels associated with each graph.

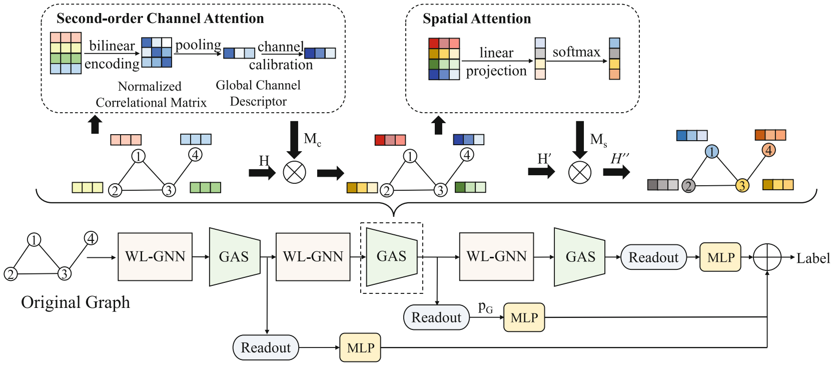


Fig. 2. The architecture of SGAN (best view in color). The WL-GNN block indicates a layer of the general WL-GNNs, which can be instantiated by GIN, 3WLGNN layers, etc. GAS block consists of a SoGCA module and a node attention module. SoGCA module uses second-order information to perform channel-wise attention, giving each dimension of hidden embeddings different weight. Node attention module obtain the importance of nodes. (Color figure online)

2.2 Graph Isomorphism Networks

Graph Isomorphism Networks [27] (GIN) is an architecture based on the Weisfeiler-Lehman Isomorphism test, which can quantify the expressive power of GNNs. GIN is as powerful as 1-WL algorithm owing to injective update and aggregation functions as:

$$\hat{\mathbf{H}}^{\ell+1} = (1+\epsilon)\,\mathbf{H}^{\ell} + \mathbf{A}\mathbf{H}^{\ell}, \tag{1}$$

$$\mathbf{H}^{\ell+1} = \text{ ReLU}\left(\text{ReLU}\left(\text{ BN}\big(\hat{\mathbf{H}}^{\ell+1}\ \mathbf{V}^{\ell}\big)\ \right)\ \mathbf{U}^{\ell}\ \right), \tag{2}$$

where ϵ can be a learnable parameter or a fixed scalar, $\mathbf{H}^{\ell} \in \mathbb{R}^{N\times d}$ is the embedding representation of all nodes derived from the ℓ^{th} layer, $\mathbf{V}^{\ell}, \mathbf{U}^{\ell} \in \mathbb{R}^{d\times d}$ are learnable matrices for layer ℓ, BN represents Batch Normalization [10], $\text{ReLU}(x) = \max(0, x)$ is the non-linear activation function, and A is the adjacency matrix of the graph.

3 Proposed Method: SGAN

In the following section, we consider the intermediate graph representation $\mathbf{H} \in \mathbb{R}^{N\times d}$ resulting from the layer of WL-GNNs, which can be instantiated by GIN, 3WLGNN layers [18], etc. Following the concept of channel in Convolutional Neural Networks (CNNs), we define each column of $\mathbf{H}$ as a **channel**.

3.1 Second-Order Global Channel Attention (SoGCA)

Existing WL-GNNs do not consider the importance of different channels for graph isomorphism test. However, for other graph-level tasks, we have analyzed the necessity of modeling feature importance in Sect. 1. Besides, recent studies in computer vision [16,17] have shown that channel interdependencies in deep CNNs are helpful for extracting more discriminative representations. Inspired by this observation, we propose a SoGCA module that incorporates channel interdependencies. As illustrated in Fig. 2, the SoGCA consists of three parts: bilinear encoding, global correlation pooling, and channel calibration.

Bilinear Encoding. The first step is to model channel interdependencies by utilizing a correlation matrix. We calculate the inner product between each pair of channels to generate a channel correlation matrix $\mathbf{D}$ as:

$$\mathbf{D} = \mathbf{H}^T\mathbf{H}, \tag{3}$$

where $\mathbf{D} \in \mathbb{R}^{d\times d}$ and each element $\mathbf{D}_{ij} = \sum_{k=1}^{N} \mathbf{H}_{ik}\mathbf{H}_{kj}$ measures the degree of second-order interdependency between two channels. A large value of $\mathbf{D}_{ij}$ indicates the i^{th} channel and the j^{th} channel are highly related.

To further improve feature representation, we normalize the channel correlation matrix $\mathbf{D}$. Following the practice in [16], we adopt signed square-root and ℓ_2 normalization, which yields

$$\mathbf{D}_{norm} = sign(\mathbf{D})\frac{\sqrt{|\mathbf{D}|}}{\|\sqrt{|\mathbf{D}|}\|_2}. \tag{4}$$

Note that the above operations are piecewise differentiable, so they can be used for end-to-end training.

Global Correlation Pooling. Then, we apply global average pooling function over the normalized correlation matrix D_{norm}. We obtain a global channel descriptor $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_d) \in \mathbb{R}^{1\times d}$ as:

$$\mathbf{z} = \frac{1}{d}\sum_{m}(\mathbf{D}_{norm})_m. \tag{5}$$

Compared with directly applying global average pooling over $\mathbf{H}$, global correlation pooling captures more useful information. Each element $\mathbf{z}_i$ encodes the second-order interdependencies between the i^{th} channel and all the other channels. So the global channel descriptor $\mathbf{z}$ can be used for learning more discriminative representation. We validate the effectiveness of global correlation pooling in Sect. 4.4.

Channel Calibration. In order to learn the attention weights and fully exploit channel interdependencies, we apply 2-layer MLPs as:

$$\mathbf{M}_c = \sigma(\text{ReLU}(\mathbf{z}\mathbf{W}_0)\mathbf{W}_1), \tag{6}$$

where $\mathbf{W}_0 \in \mathbb{R}^{d\times r}$, $\mathbf{W}_1 \in \mathbb{R}^{r\times d}$ are learnable weights, r is a hyper-parameter which controls capacity of the attention module. Finally, we obtain the channel attention map $\mathbf{M}_c$ to rescale the graph representation as:

$$\mathbf{H}' = \mathbf{M}_c \otimes \mathbf{H}. \tag{7}$$

3.2 Node Attention Module

Apart from channel attention, node attention is also important for extracting informative signals. For example, in molecular chemistry, the functional groups are usually related to a lot of chemical properties, while many other nodes do not influence the properties. So we also generate a node attention map, which can focus on important nodes and provides complementary information to channel attention.

If we apply the above bilinear encoding to calculate node attention, we would get a node correlation matrix of $\mathbb{R}^{N \times N}$. As a result, it would be intractable to handle this matrix in large graphs. For simplicity, we use a linear projection followed by a softmax function to calculate the node attention score as:

$$\mathbf{M}_n = softmax(\sigma(\mathbf{H}'\mathbf{W}_2)), \tag{8}$$

where $\mathbf{W}_2 \in \mathbb{R}^{d \times 1}$ are the trainable weight matrix for node attention, $softmax(\mathbf{x}) = e^{\mathbf{x}}/\sum e^{\mathbf{x}}$ is used for normalization.

After that, $\mathbf{M}_n$ is applied to obtain the refined graph representation $\mathbf{H}''$, which is fed into the follow-up layers of WL-GNNs:

$$\mathbf{H}'' = \mathbf{M}_n \otimes \mathbf{H}'. \tag{9}$$

In fact, the above second-order channel attention resemble that in SOPOOL [23]. However, SGAN is distinct from SOPOOL in both motivation and technique. On the one hand, SGAN points out the limitations of directly applying WL-GNNs to graph classification and graph regression tasks, while SOPOOL only focuses on strenthen important features. On the other hand, SGAN also considers the node importance via node attention.

4 Experiments

Datasets. For a comprehensive evaluation of our proposed method, we use eight benchmark datasets in benchmark-GNNs [4] and OGB [9] for graph analytical tasks, including classification, regression, and graph isomorphism test. For graph classification task, two widely used protein datasets [12], ENZYMES and DD are used. We also conduct experiments on larger datasets of MNIST and CIFAR10. These two datasets convert the original images into graphs using super-pixels. For graph regression task, we use a subset of ZINC molecular graphs dataset [11] to regress the constrained solubility of a molecule. We also apply another two molecular graphs, OGBG-molesol and OGBG-molfreesolv in OGB [9] dataset for regression. Furthermore, we use the Circular Skip Link (CSL) dataset [21] for graph isomorphism test. The statistics of these datasets are summarized in Table 1.

4.1 Experimental Setup

Compared Methods. We compare SGAN with four widely used message passing-based GNNs: GCN [14], GraphSage [6], GAT [22] and MoNet [19]. For attention-guided GNNs, we compare with SAGPool [15] and cGAO [5]. For WL-GNNs, we select GIN [27], 3WLGNN [18] and PNA [2] as baselines.

Implementation Details. We closely follow benchmark-GNNs to set hyper-parameters. We perform grid-search to select the initial learning rate from a range of $1e^{-3}$ to $7e^{-5}$. The learning rate decay factor is 0.5 and the model is optimized with Adam [13] optimizer. We use classification accuracy as evaluation metric for all datasets except ZINC. For the regression task on ZINC, we measure the performance by using Mean Absolute Error (MAE). We report the average results of MNIST, CIFAR10 and ZINC over 4 runs with 4 different seeds. The results on CSL dataset are obtained by running 20 times with different seeds. All baselines on all benchmark-GNNs datasets are trained with a budget of 100k parameters. Following experimental protocols of OGB, we use edge features and report the Root Mean Squared Error (RMSE). Notably, we implement our baselines on Huawei Mindspore platform.

4.2 Performance Comparison

We compare the performance of SGAN with baseline methods on four graph classification datasets and one graph regression dataset. The results are shown in Table 2. We find that:

- **Traditional WL-GNNs perform relatively poor.** Although provably powerful in terms of graph isomorphism test, GIN and 3WLGNN do not outperform GCN or GAT obviously. This indicates that graph isomorphism property is not sufficient to yield satisfactory results in graph classification and regression tasks.

Table 1. Statistics of all datasets in experiments.

Dataset	#Graphs	#Classes	Tasks
ENZYMES	600	6	Classification
DD	1178	2	Classification
MNIST	70k	10	Classification
CIFAR10	60k	10	Classification
ZINC	12k	–	Regression
OGBG-molesol	1128	–	Regression
OGBG-molfreesolv	642	–	Regression
CSL	150	10	Isomorphism

Table 2. Results of graph classification/regression with the best performances highlighted in bold. Classification accuracies (%) are reported for all datasets except ZINC. ↓ indicates lower is better for the regression loss. OOM represents out of memory.

Method	ENZYMES	DD	MNIST	CIFAR10	ZINC (↓)
GCN	65.833	72.758	90.705	55.710	0.459
GraphSage	65.000	73.433	97.312	65.767	0.468
MoNet	63.000	71.736	90.805	54.655	0.397
GAT	**68.500**	75.900	95.535	64.223	0.475
SAGPool	66.833	75.354	92.375	57.032	0.425
cGAO	63.833	73.685	91.833	54.824	0.468
GIN	65.333	71.910	96.485	55.255	0.387
3WLGNN	61.000	OOM	95.075	59.175	0.407
PNA	–	–	97.190	**70.210**	**0.320**
SGAN(GIN)	**68.333**	**78.010**	**97.500**	58.750	**0.267**
SGAN(3WLGNN)	62.000	OOM	96.212	63.125	**0.384**
SGAN(PNA)	–	–	**97.650**	**70.340**	**0.249**

Table 3. Ablation study of SGAN(GIN) on benchmark-GNNs datasets. We investigate the effectiveness of SoGCA and node attention (NA), respectively. The first line is the results of GIN.

SoGCA	NA	ENZYMES	DD	MNIST	CIFAR10	ZINC (↓)	Isomorphism test
×	×	65.333	71.910	96.485	55.255	0.387	99.333
×	√	65.167	77.428	97.205	58.403	0.272	99.333
√	×	67.500	72.147	97.410	58.126	0.291	99.333
√	√	**68.000**	**78.010**	**97.500**	**58.750**	**0.267**	**99.333**

- **Global attention mechanisms improves the performance.** SAGPool achieves better performance than GCN in most cases. We attribute this improvement to capturing important nodes. However, neither SAGPool nor cGAO can obtain state-of-the-art results, because they only consider one kind of attention. Besides, since both SAGPool and cGAO employ GCN as base neighborhood aggregator, their performances may also be dragged down by GCN. Furthermore, cGAO may even degrades the performance on some datasets, which might be caused by lacking learnable parameters in attention process.
- **SGAN consistently achieves the best performance on all datasets.** SGAN improves the performance over GIN and 3WLGNN by remarkable margins, which verifies the necessity of modeling graph-level channel attention and node attention. Specifically, even compared with the very recent state-of-the-art PNA method, the proposed SGAN yields better results. The reason

is that PNA only focuses on capturing the **local** neighborhood distributions, while SGAN(PNA) can extract additional important **global** information.

4.3 Ablation Study of Attention Modules

In this subsection, we investigate the contributions of SoGCA and node attention module (NA) to the performance. We conduct experiments based on SGAN(GIN) by removing all SoGCA modules and NA modules, respectively. The results are shown in Table 3 and Table 4. We have the following observations:

Table 4. Ablation study of SGAN(GIN) on OGB datasets for regression. The first line is the results of GIN.

SoGCA	NA	OGBG-molesol (↓)	OGBG-molfreesolv (↓)
×	×	0.998	2.151
×	√	1.087	2.709
√	×	**0.875**	**1.872**
√	√	1.028	2.424

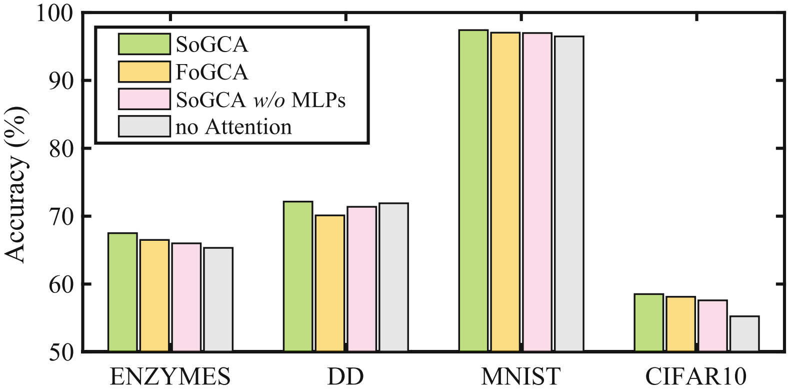


Fig. 3. The performance of different channel attention variants on graph classification. We plot the classification accuracy for the four datasets. Higher are better for these histograms. w/o is the abbreviation of without.

- **The characteristics of datasets influence the performance on different modules.** Interestingly, SoGCA brings more improvement on ENZYMES while NA performs better on DD and ZINC dataset. We guess these discrepancies are due to different characteristics of datasets. For instance, the input node features contain continuous values ranging from −10 to around 300 in ENZYMES. In contrast, the input node attributes in DD and ZINC dataset follow one-hot encoding, where most of the numbers in attribute vectors equal to zero. As a result, SoGCA performs better on ENZYMES because channel interdependencies might be more important to this dataset.

- **Compared to NA, the proposed SoGCA achieves more stable improvements in most cases.** Applying NA decreases the performance on some datasets, such as ENZYMES in Table 3 and the two OGB datasets in Table 4. We conjecture that most of the nodes in these datasets are important and NA neglects this information. By contrast, SoGCA still has satisfactory improvements in these datasets, which therefore verifies the necessity of modeling channel attention.
- **The combination of SoGCA and NA usually achieves better performance.** Except for the situation when NA fails, the combination of these two attention modules usually produces better performance. Since these two modules focuses on different aspects of node embedding, they provide complementary information to each other. Therefore, it is reasonable to combine them.
- **SGAN can retain the expressive power of vanilla WL-GNNs.** We conduct experiments on graph isomorphism task to study the expressive power of SGAN(GIN). For isomorphism test, it is required to strictly distinguishing these two graphs. Therefore, it might be natural that attention modules degrade the performance for graph isomorphism task. However, we find that removing SoGCA or NA makes no difference to the performance. This demonstrates that SoGCA and NA can learn adaptively according to the task. When the attention weights are near uniform distribution, SGAN(GIN) degenerates to GIN and keeps the same accuracy as GIN.

4.4 Study on Channel Interdependencies

In this subsection, we make a deeper study on the effect of modeling channel interdependencies. As presented in Sect. 3.1, our SoGCA module consists of three parts: Bilinear Encoding, Global Correlation Pooling (GCP) and Channel Calibration. Both of the first two parts and the third part can model channel interdependencies. We remove these parts respectively, yielding two model variants. The first variant is called First-order Graph Channel Attention method (**FoGCA**). The second variant is abbreviated as **SoGCA *w/o* MLPs**, which removes the 2-layer MLPs and uses the global channel descriptor $\mathbf{z}$ as channel attention map. We compare the results of these models based on GIN (i.e., no attention variant), which are shown in Fig. 3. It can be found that modeling channel attention can generally improve the performance of GIN. Either removing second-order modeling or MLPs degrades the overall performance, which verifies the necessity of modeling channel interdependencies and channel attention.

5 Conclusion

In this paper, we have proposed a novel second-order global attention networks for graph classification and regression tasks. The key of SGAN is the newly proposed SoGCA layer, which can capture second-order channel interdependencies and highlight important information. Compared with other previous WL-GNNs which focus on graph isomorphism property, our proposed SGAN can highlight determinative information from both channel level and node level. Comprehensive experiments have demonstrated the rationality and necessity of modeling channel attention and capturing second-order statistics of features for GNNs.

Acknowledgements. This work is jointly supported by National Natural Science Foundation of China (62141608, U19B2038) and CAAI Huawei MindSpore Open Fund.

References

1. Chen, F., Chen, X., Meng, F., Li, P., Zhou, J.: GoG: relation-aware graph-over-graph network for visual dialog. arXiv preprint arXiv:2109.08475 (2021)
2. Corso, G., Cavalleri, L., Beaini, D., Liò, P., Veličković, P.: Principal neighbourhood aggregation for graph nets. In: Advances in Neural Information Processing Systems (2020)
3. Cui, Z., et al.: DyGCN: dynamic graph embedding with graph convolutional network. IEEE Trans. Neural Netw. Learn. Syst. (2022)
4. Dwivedi, V.P., Joshi, C.K., Laurent, T., Bengio, Y., Bresson, X.: Benchmarking graph neural networks. arXiv preprint arXiv:2003.00982 (2020)
5. Gao, H., Ji, S.: Graph representation learning via hard and channel-wise attention networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
6. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems (2017)
7. Hu, F., Liping, W., Qiang, L., Wu, S., Wang, L., Tan, T.: GraphDIVE: graph classifcation by mixture of diverse experts. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence (2022)
8. Hu, F., Zhu, Y., Wu, S., Huang, W., Wang, L., Tan, T.: GraphAIR: graph representation learning with neighborhood aggregation and interaction. Pattern Recogn. **112**, 107745 (2021)
9. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687 (2020)
10. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning (2015)
11. Irwin, J.J., Sterling, T., Mysinger, M.M., Bolstad, E.S., Coleman, R.G.: ZINC: a free tool to discover chemistry for biology. J. Chem. Inf. Model. **52**, 1757–1768 (2012)
12. Kersting, K., Kriege, N.M., Morris, C., Mutzel, P., Neumann, M.: Benchmark data sets for graph kernels (2016)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
15. Lee, J., Lee, I., Kang, J.: Self-attention graph pooling. In: Proceedings of the 36th International Conference on Machine Learning (2019)
16. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: Proceedings of the IEEE International Conference on Computer Vision (2017)

17. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. In: Proceedings of the IEEE International Conference on Computer Vision (2015)
18. Maron, H., Ben-Hamu, H., Serviansky, H., Lipman, Y.: Provably powerful graph networks. In: Advances in Neural Information Processing Systems (2019)
19. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
20. Morris, C., et al.: Weisfeiler and Leman go neural: higher-order graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
21. Murphy, R.L., Srinivasan, B., Rao, V., Ribeiro, B.: Relational pooling for graph representations. In: Proceedings of the 36th International Conference on Machine Learning (2019)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations (2018)
23. Wang, Z., Ji, S.: Second-order pooling for graph neural networks. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
24. Weisfeiler, B., Lehman, A.A.: A reduction of a graph to a canonical form and an algebra arising during this reduction. Nauchno-Technicheskaya Informatsia (1968)
25. Wu, J., Liu, Q., Xu, W., Wu, S.: Bias mitigation for evidence-aware fake news detection by causal intervention. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (2022)
26. Xia, B.N., Gong, Y., Zhang, Y., Poellabauer, C.: Second-order non-local attention networks for person re-identification. In: Proceedings of the IEEE International Conference on Computer Vision (2019)
27. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2019)
28. Xu, W., Wu, J., Liu, Q., Wu, S., Wang, L.: Evidence-aware fake news detection with graph neural networks. In: Proceedings of the ACM Web Conference 2022 (2022)
29. Zhang, D., Chen, X., Xu, S., Xu, B.: Knowledge aware emotion recognition in textual conversations via multi-task incremental transformer. In: Proceedings of the 28th International Conference on Computational Linguistics (2020)
30. Zhang, M., Wu, S., Gao, M., Jiang, X., Xu, K., Wang, L.: Personalized graph neural networks with attention mechanism for session-aware recommendation. IEEE Trans. Knowl. Data Eng. **34**, 3946–3957 (2022)
31. Zhang, M., Wu, S., Yu, X., Liu, Q., Wang, L.: Dynamic graph neural networks for sequential recommendation. IEEE Trans. Knowl. Data Eng. (2022)
32. Zhang, Y., Yu, X., Cui, Z., Wu, S., Wen, Z., Wang, L.: Every document owns its structure: inductive text classification via graph neural networks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)