Appendix: Kernel Readout for Graph Neural Networks

Jiajun Yu[1,3,5*], **Zhihao Wu**[2,3*], **Jinyu Cai**[4], **Adele Lu Jia**[1†] and **Jicong Fan**[2,3†]
[1]College of Information and Electrical Engineering, China Agricultural University, Beijing, China
[2]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[3]Shenzhen Research Institute of Big Data, Shenzhen, China
[4]Institute of Data Science, National University of Singapore, Singapore
[5]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
2017307070311@cau.edu.cn, zhihaowu1999@gmail.com, jinyucai1995@gmail.com, Ljia@cau.edu.cn, fanjicong@cuhk.edu.cn

Abstract

In this appendix, we first present the proof for Theorem 1. Then we detail the experimental settings and results. Specifically, Section B.1, B.2, and B.3 present more details about the experimental settings, various datasets, compared methods, and hyperparameters. Section B.4 visualizes the graph embedding on the DD dataset.

A Proof for Theorem 1

Proof. Let the given graph be $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and $\mathbf{P}$ is an arbitrary permutation matrix, the permuted graph is denoted as $\mathcal{G}' = (\mathbf{PAP}^\top, \mathbf{PX})$. We recall that the graph learning model $\mathrm{M}_{\Upsilon,\Psi,\mathcal{W}}$ contains a feature permutation invariant kernel k (e.g. Gaussian, Polynomial and Laplace kernel), and it can be formulated as

$$\mathrm{M}_{\Upsilon,\Psi,\mathcal{W}}(\mathcal{G}) = h_\Upsilon\left(\phi\Big(g_\Psi\big(f^{\mathrm{GNN}}_{\mathcal{W}}(\mathcal{G})\big)\Big)^\top \phi\Big(f^{\mathrm{GNN}}_{\mathcal{W}}(\mathcal{G})\Big)\right), \quad (1)$$

where h_Υ denotes the fusion model, g_Ψ is the center learning model and ϕ is a mapping induced by kernel k. Note that $f^{\mathrm{GNN}}_{\mathcal{W}}$ is a GNN with permutation invariant aggregation, i.e., for node representation $\mathbf{H}_{\mathcal{G}'} = f^{\mathrm{GNN}}_{\mathcal{W}}(\mathcal{G}')$, we have

$$\mathbf{H}_{\mathcal{G}'} = \mathbf{P}\mathbf{H}_{\mathcal{G}}. \quad (2)$$

which leads to

$$\begin{aligned}\mathbf{C}_{\mathcal{G}'} &= g_\Psi(\mathbf{H}_{\mathcal{G}'}) = g_\Psi(\mathbf{P}\mathbf{H}_{\mathcal{G}}) \\ &= [s_1 \cdot f_\Theta(\mathbf{P}\mathbf{H}_{\mathcal{G}}), \cdots, s_m \cdot f_\Theta(\mathbf{P}\mathbf{H}_{\mathcal{G}})].\end{aligned} \quad (3)$$

Since f_Θ is performed on each row of $\mathbf{H}_{\mathcal{G}}$ separately, it is obvious that

$$f_\Theta(\mathbf{P}\mathbf{H}_{\mathcal{G}}) = \mathbf{P} f_\Theta(\mathbf{H}_{\mathcal{G}}), \quad (4)$$

which means

$$\mathbf{C}_{\mathcal{G}'} = \mathbf{P}\mathbf{C}_{\mathcal{G}}. \quad (5)$$

Given the feature permutation invariant kernel k and its inducing mapping ϕ, for $\mathrm{M}_{\Upsilon,\Psi,\mathcal{W}}(\mathcal{G}')$, we have

$$\begin{aligned}\mathrm{M}_{\Upsilon,\Psi,\mathcal{W}}(\mathcal{G}') &= h_\Upsilon\big(\phi(\mathbf{P}\mathbf{C}_{\mathcal{G}})^\top \phi\big(\mathbf{P}\mathbf{H}_{\mathcal{G}}\big)\big) \\ &= h_\Upsilon\Big(\big[k(\mathbf{P}\mathbf{C}_{\mathcal{G}}[;,i], \mathbf{P}\mathbf{H}_{\mathcal{G}}[:,j])\big]_{m\times d}\Big) \\ &= h_\Upsilon\Big(\big[k(\mathbf{C}_{\mathcal{G}}[;,i], \mathbf{H}_{\mathcal{G}}[:,j])\big]_{m\times d}\Big) \\ &= h_\Upsilon\big(\phi(\mathbf{C}_{\mathcal{G}})^\top \phi\big(\mathbf{H}_{\mathcal{G}}\big)\big) \\ &= \mathrm{M}_{\Upsilon,\Psi,\mathcal{W}}(\mathcal{G}),\end{aligned} \quad (6)$$

which completes the proof. □

B Detailed Experimental Settings and Results

B.1 Dataset

In this article, we utilize eight graph-level datasets. Among the six chemical molecule datasets, node representations are encoded as one-hot vectors. Conversely, the two social network graph datasets lack initial feature vectors, prompting us to uniformly assign them a value of 1. The statistical overview of these datasets is presented in Table 1. Subsequently, we provide detailed descriptions of each dataset in the following section.

- MUTAG dataset comprises 188 chemical compounds categorized into two classes based on their mutagenic impact on bacteria. The chemical data, sourced from http://cdb.ics.uci.edu, was transformed into graphs wherein vertices denote atoms and edges signify chemical bonds. Hydrogen atoms were explicitly excluded, and vertices were annotated with atom types, while edges were labeled with bond types (single, double, triple, or aromatic).
- DD is a dataset featuring 1178 protein structures (Dobson and Doig, 2003). Each protein is depicted as a graph, where nodes represent amino acids, and edges connect two nodes if they are within a 6 Angstrom proximity. The predictive objective involves classifying protein structures into enzymes and non-enzymes.
- PROTEINS dataset in bioinformatics comprises nodes that represent secondary structure elements, each assigned one of three distinct labels. An edge connecting

*Jiajun Yu and Zhihao Wu contributed equally to this work.
†Corresponding authors.

Datasets	#Graphs	Avg.#Nodes	Avg.#Edges	# Features	#Classes
MUTAG	188	17.93	19.79	7	2
DD	1,178	284.32	715.66	89	2
PROTEINS	1,113	39.06	72.82	3	2
Mutagen	4,337	30.32	30.77	14	2
NCI1	4,110	29.87	32.30	37	2
IMDB-B	1,000	19.77	96.53	None	2
IMDB-M	1,500	13.00	65.94	None	3
OGBG-Molhiv	41,127	25.5	27.5	9	2

Table 1: The statistics of datasets.

two nodes signifies their adjacency either in the amino acid sequence or within the three-dimensional spatial arrangement.

- NCI1 represents two balanced subsets of datasets containing chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines, respectively.
- Mutagenicity constitutes a dataset of chemical compounds, particularly drugs, classified into two categories: mutagen and non-mutagen.
- IMDB-B and IMDB-M are datasets about collaborative efforts in the film industry. In these datasets, each graph symbolizes an ego network for individual actors/actresses. Nodes correspond to actors/actresses, and edges delineate co-occurrences in movies. Classification of the graphs is based on the genre of the associated movies.
- OGBG-Molhiv constitutes a molecular dataset wherein each graph serves as a structural representation of an individual molecule. Nodes within these graphs denote atoms, while edges depict chemical bonds. The input node features are encapsulated within a nine-dimensional array, encompassing attributes such as atomic number and chirality. Additionally, supplementary atom features are incorporated, including formal charge and the determination of the atom's involvement in a ring structure.

B.2 Baseline and BackBone

We use a total of 8 backbones to encode the graph to verify KerRead's generalization. They were divided into six supervised backbones and two unsupervised backbones. We will introduce these backbones one by one.

Supervised methods

- GCN [Welling and Kipf, 2017] simplifies the Chebyshev polynomial to the first order and introduces a straightforward yet highly efficient graph convolution method.
- GraphSAGE [Hamilton *et al.*, 2017] operates by sampling and aggregating information from the neighborhood of a node. It allows for scalable and efficient learning by aggregating information from a node's local subgraph.
- GAT [Veličković *et al.*, 2018] uses attention mechanisms to assign different weights to the neighbors of a node, allowing it to focus on more relevant nodes during aggregation.
- SGC [Wu *et al.*, 2019] simplifies the graph convolution operation by removing non-linearities and activation functions.
- GIN [Chen *et al.*, 2019] uses a learnable aggregation function to iteratively update node representations and achieve graph isomorphism invariance.
- GUNet [Gao and Ji, 2019] is a variant of the U-Net architecture adapted for graph-structured data.

Unsupervised methods

- InfoGraph [Sun *et al.*, 2019] emphasis lies in the optimization of mutual information between local and global structural attributes within a graph.
- GraphCL [You *et al.*, 2020] aims to learn node representations by contrasting positive pairs (augmented views of the same graph) against negative pairs (views from different graphs).

Additionally, this paper presents a baseline approach. Simple readout methods, such as Sum, Max, and Mean are straightforward and require no detailed explanation. Set2set [Vinyals *et al.*, 2016] utilizes a Long Short-Term Memory (LSTM) network [Hochreiter and Schmidhuber, 1997] to transform the node embedding set into a graph vector. Attention [Li *et al.*, 2016] employs an attention function to generate attention scores that are then used for a weighted sum of the node embedding values. Deep Sets [Zaheer *et al.*, 2017] propose a deep network capable of operating on sets of arbitrary sizes. SRead [Lee *et al.*, 2021] introduces learnable structural prototypes to capture graph semantics relevant to the global structure. Set Transformer [Buterez *et al.*, 2022] incorporates encoder-decoder modules into the readout process. Janossy MLP and Janossy GRU [Buterez *et al.*, 2022] employ the Gated Recurrent Unit (GRU) network [Chung *et al.*, 2014] and Multi-Layer Perceptron (MLP) as aggregators, combining the Janossy pooling idea to obtain the final node embedding.

B.3 Parameter Setting

For the graph classification task, the training, validation, and test set proportions are set at 8:1:1. We perform 10-fold cross-

Parameter Name	Setting
Optimizer	Adam [Kingma and Ba, 2015]
Max training epochs	300
Learning rate	1e-3
Dimension of hidden layer	300
Weight decay	1e-4
Batchsize	128
Dropout ratio	0.5
# Graph convolution layers	2

Table 2: Parameter setting.

validation and record the average accuracy and standard deviation for MUTAG, DD, PROTEINS, NCI1, Mutagenicity, IMDB-B, and IMDB-M. For OGBG-Molhiv, due to its class imbalance, we adhere to the approach outlined in [Hu *et al.*, 2020] and opt for the AUC as the evaluation metric. For the graph clustering task, we employ Accuracy (ACC) and Normalized Mutual Information (NMI) [Estévez *et al.*, 2009] as the evaluation metrics. And K-means [Lloyd, 1982] are employed to generate the clustering results.

The experiments are conducted on an RTX A5000 GPU (24GB), a CPU equipped with 16 vCPUs (Intel(R) Xeon(R) Platinum 8350C CPU @ 2.60GHz), and 42GB of memory. The setting of general parameters is shown in Table 2, and for the KerRead, we employed the Gaussian kernel and used 4 adaptive centers to learn the graph embeddings.

B.4 Experiments of Visualization

In this study, we employed the t-SNE algorithm [Van der Maaten and Hinton, 2008] to visualize the graph embeddings obtained from KerRead and 10 other readout functions on the DD dataset. InfoGraph was utilized as the GNN backbone for representation learning and the visualization is depicted in Figure 1. Overall, readout functions exhibited a significant impact on graph representation learning. For instance, GRU demonstrated suboptimal performance on the DD dataset, leading to complete overlap in its generated graph embeddings. Furthermore, the visualization shows that KerRead outperforms other readout functions, with minimal overlap observed between the two categories.

References

[Buterez *et al.*, 2022] David Buterez, Jon Paul Janet, Steven J Kiddle, Dino Oglic, and Pietro Liò. Graph neural networks with adaptive readouts. In *NeurIPS*, pages 19746–19758, 2022.

[Chen *et al.*, 2019] Ting Chen, Song Bian, and Yizhou Sun. Are powerful graph neural nets necessary? a dissection on graph classification. *arXiv preprint arXiv:1905.04579*, 2019.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS Workshop on Deep Learning*, 2014.

[Estévez *et al.*, 2009] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[Gao and Ji, 2019] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *ICML*, pages 2083–2092, 2019.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, pages 22118–22133, 2020.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Lee *et al.*, 2021] Dongha Lee, Su Kim, Seonghyeon Lee, Chanyoung Park, and Hwanjo Yu. Learnable structural semantic readout for graph classification. In *ICDM*, pages 1180–1185, 2021.

[Li *et al.*, 2016] Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. Gated graph sequence neural networks. In *ICLR*, 2016.

[Lloyd, 1982] Stuart Lloyd. Least squares quantization in pcm. *IEEE T-IT*, 28(2):129–137, 1982.

[Sun *et al.*, 2019] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*, 2019.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Vinyals *et al.*, 2016] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In *ICLR*, 2016.

[Welling and Kipf, 2017] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Wu *et al.*, 2019] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, pages 5812–5823, 2020.

[Zaheer *et al.*, 2017] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *NeurIPS*, 2017.

(a) Sum (b) Max (c) Mean

(d) Set2Set (e) Attention (f) Deep Sets

(g) SRead (h) Set Transformer (i) Janossy MLP

(j) Janossy GRU (k) KerRead

Figure 1: 2D visualization of graph embeddings on the DD dataset. Graph embeddings are obtained by InfoGraph.