Appendix for the Paper: “SizeShiftReg: a Regularization Method for Improving Size-Generalization in Graph Neural Networks”

Davide Buffelli
Department of Information Engineering
University of Padova
Padova, Italy 35131
davide.buffelli@unipd.it

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
Cambridge, United Kingdom CB3 0FD
pietro.lio@cl.cam.ac.uk

Fabio Vandin
Department of Information Engineering
University of Padova
Padova, Italy 35131
fabio.vandin@unipd.it

Appendix

This Appendix contains the following additional material. In Section A we report more statistics regarding the considered datasets. In Section B we report the results for the CKA analysis on the other datasets not shown in the main paper for space limitations. In Section C we report more details on the hyperparameters of the models used for our experiments. In Section D we report the results for the ablation study on the other models not shown in the main paper for space limitations. In Section E we show the overhead (in terms of training time) incurred by our method. In Section F we report information on the computing resources used to perform our experimental results, links to access the datasets, and the licence of the publicly available libraries used in our code.

A Dataset Information

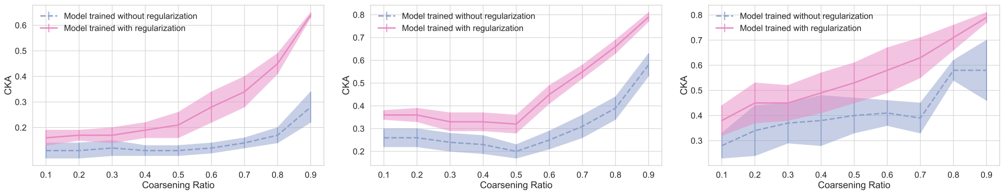
Table 1: Dataset statistics, this table is taken from Yehudai et al. [13], Bevilacqua et al. [1].

	NCI1			NCI109		
	ALL	SMALLEST 50%	LARGEST 10%	ALL	SMALLEST 50%	LARGEST 10%
CLASS A	49.95%	62.30%	19.17%	49.62%	62.04%	21.37%
CLASS B	50.04%	37.69%	80.82%	50.37%	37.95%	78.62%
NUM OF GRAPHS	4110	2157	412	4127	2079	421
AVG GRAPH SIZE	29	20	61	29	20	61

	PROTEINS			DD		
	ALL	SMALLEST 50%	LARGEST 10%	ALL	SMALLEST 50%	LARGEST 10%
CLASS A	59.56%	41.97%	90.17%	58.65%	35.47%	79.66%
CLASS B	40.43%	58.02%	9.82%	41.34%	64.52%	20.33%
NUM OF GRAPHS	1113	567	112	1178	592	118
AVG GRAPH SIZE	39	15	138	284	144	746

In Table 1 (taken from [13, 1]) we report the information about the graphs in the considered datasets.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).



(a) NCI1 (b) NCI109 (c) PROTEINS

Figure 1: Average CKA values between the node embeddings generated by the same model for the original graphs in (a) NCI1, (b) NCI109, (c) PROTEINS, and their coarsened versions. A model trained with our regularization produces representations of the coarsened graphs that are more aligned to the representations produced for the original graphs (i.e. it learns to be robust to size shifts).

B Embedding Analysis Results for Other Datasets

In Figure 1 we show the continuation of Figure 2 (b) from the main paper, including plots on all datasets. In more detail, we report plots for the average CKA [7] values between the node embeddings for the original graphs and the node embeddings for the coarsened versions of the graphs, both generated by the same model, for different ratios. The plot shows that a model trained with our regularization is learning to be more robust to size shifts, as its representation for coarsened versions of the graphs are much more aligned (higher CKA values) to the representations for the original graphs compared to a model trained without regularization. This is even more apparent for low coarsening ratios (i.e., when the shift is stronger), indicating that our regularization is in fact making the model more robust to size shifts. The reason why the CKA values for a model trained with and without regularization tend to become similar for small ratios (which is instead not apparent on the DD dataset shown in Figure 2 (b) in the main paper) is that the graphs in NCI1, NCI109, and PROTEINS are much smaller than the graphs in DD (see Section A), and hence graphs tend to become completely uninformative with ratios lower than 0.5.

C Hyperparameters and Evaluation Procedure

Hyperparameters. As we follow exactly the same procedures as previous work [13, 1], we refer the reader to those papers for an in depth discussion, and we report below the description of the standard GNN models used in our experiments (which follows from the above mentioned papers).

The GCN[6], GIN [12], and PNA [2] models used in our experiments have 3 graph convolution layers and a final multi-layer perceptron with a softmax activation function to obtain the predictions. Batch norm is used in between graph convolution layers, and ReLU is used as activation function. The networks are trained with a dropout of 0.3, and were tuned using the validation set. In particular the batch size was chosen between 64 and 128, the learning rate between $0.01, 0.005$, and 0.001, and the network width between 32 and 64. All models are trained with early stopping (taking the wights related to the epoch with the highest MCC on the validation set), and the different classes were weighted in the supervised loss function according to the frequency of a class in the training data (to deal with the imbalance in the data, as explained in the main paper).

When applying our method in Section 3 and in Table 2 and Table 3 in the main paper, we use $\lambda = 0.1$ and $C = (0.8, 0.9)$ for all models and datasets. These parameters were taken from a tuning procedure on the validation set described below.

Tuning and Evaluation Procedure. To identify the values of λ and C to use for our method, we tried different values ($\lambda = \{1.0, 0.1, 0.01, 0.001\}$ and $C = \{(0.5), (0.8), (0.9), (0.5, 0.8), (0.5, 0.9), (0.8, 0.9)\}$) on the validation set using a GIN model on the PROTEINS dataset. We found that $\lambda = 0.1$ and $C = (0.8, 0.9)$ performed best on the validation set. As the results on the validation sets were good (i.e., they were leading to better performance with respect to a model trained without regularization), we adopted the same values of λ and C for all datasets and models to show that our method can work without extensive (and expensive)

Table 2: Table shows mean (standard deviation) Matthews correlation coefficient (MCC) for a GCN model trained with our proposed regularization with the Spectral Graph Coarsening (SGC) strategy applied with different coarsening ratios.

Datasets Ratio	**NCI1**	**NCI109**	**PROTEINS**	**DD**
0.1	0.09 ± 0.12	0.23 ± 0.07	0.05 ± 0.14	0.15 ± 0.09
0.2	0.17 ± 0.11	0.23 ± 0.07	0.05 ± 0.16	0.20 ± 0.08
0.3	0.18 ± 0.10	0.23 ± 0.07	0.08 ± 0.18	0.23 ± 0.07
0.4	0.21 ± 0.07	0.24 ± 0.06	0.09 ± 0.17	0.24 ± 0.09
0.5	0.23 ± 0.07	0.22 ± 0.05	0.21 ± 0.14	0.27 ± 0.09
0.6	0.24 ± 0.05	0.21 ± 0.06	0.23 ± 0.17	0.25 ± 0.08
0.7	0.26 ± 0.06	0.21 ± 0.05	0.25 ± 0.16	0.24 ± 0.08
0.8	0.27 ± 0.06	0.20 ± 0.04	0.29 ± 0.15	0.27 ± 0.08
0.9	0.25 ± 0.07	0.19 ± 0.05	0.22 ± 0.14	0.25 ± 0.09
0.1-0.9	0.15 ± 0.11	0.21 ± 0.06	0.09 ± 0.15	0.21 ± 0.09
0.5-0.9	0.22 ± 0.05	0.22 ± 0.06	0.28 ± 0.12	0.25 ± 0.08
0.8-0.9	0.25 ± 0.06	0.19 ± 0.06	0.29 ± 0.13	0.26 ± 0.07
0.3-0.7	0.21 ± 0.07	0.22 ± 0.05	0.10 ± 0.14	0.23 ± 0.07
ALL	0.17 ± 0.08	0.23 ± 0.09	0.12 ± 0.16	0.21 ± 0.09

Table 3: Table shows mean (standard deviation) Matthews correlation coefficient (MCC) for a GIN model trained with our proposed regularization with the Spectral Graph Coarsening (SGC) strategy applied with different coarsening ratios.

Datasets Ratio	**NCI1**	**NCI109**	**PROTEINS**	**DD**
0.1	0.07 ± 0.11	0.04 ± 0.11	0.26 ± 0.12	0.22 ± 0.10
0.2	0.07 ± 0.11	0.07 ± 0.11	0.30 ± 0.10	0.24 ± 0.12
0.3	0.07 ± 0.11	0.10 ± 0.09	0.29 ± 0.15	0.25 ± 0.09
0.4	0.02 ± 0.13	0.09 ± 0.09	0.31 ± 0.15	0.24 ± 0.09
0.5	0.07 ± 0.11	0.10 ± 0.11	0.32 ± 0.12	0.25 ± 0.09
0.6	0.02 ± 0.11	0.08 ± 0.07	0.34 ± 0.12	0.26 ± 0.10
0.7	0.01 ± 0.09	0.09 ± 0.06	0.31 ± 0.14	0.25 ± 0.11
0.8	0.15 ± 0.10	0.15 ± 0.07	0.36 ± 0.11	0.27 ± 0.08
0.9	0.22 ± 0.08	0.19 ± 0.05	0.34 ± 0.10	0.25 ± 0.10
0.1-0.9	0.09 ± 0.10	0.05 ± 0.09	0.32 ± 0.15	0.22 ± 0.10
0.5-0.9	0.03 ± 0.10	0.09 ± 0.07	0.32 ± 0.13	0.23 ± 0.09
0.8-0.9	0.23 ± 0.08	0.20 ± 0.05	0.36 ± 0.11	0.25 ± 0.09
0.3-0.7	0.05 ± 0.12	0.08 ± 0.09	0.26 ± 0.15	0.24 ± 0.10
ALL	0.05 ± 0.12	0.11 ± 0.09	0.25 ± 0.16	0.23 ± 0.10

Table 4: Table shows mean (standard deviation) Matthews correlation coefficient (MCC) for a GCN model trained with our proposed regularization and different coarsening strategies: Spectral Clustering (SC), Spectral Graph Coarsening (SGC), Multilevel Graph Coarsening (MLGC), K-Means (KMEANS).

Datasets	**NCI1**	**NCI109**	**PROTEINS**	**DD**
GCN-SGC	0.25 ± 0.06	0.19 ± 0.06	0.29 ± 0.13	0.26 ± 0.07
GCN-MLGC	0.21 ± 0.07	0.21 ± 0.06	0.27 ± 0.14	0.25 ± 0.06
GCN-SC	0.28 ± 0.07	0.18 ± 0.06	0.24 ± 0.16	0.24 ± 0.07
GCN-KMEANS	0.28 ± 0.06	0.18 ± 0.05	0.25 ± 0.15	0.24 ± 0.07

Table 5: Table shows mean (standard deviation) Matthews correlation coefficient (MCC) for a GIN model trained with our proposed regularization and different coarsening strategies: Spectral Clustering (SC), Spectral Graph Coarsening (SGC), Multilevel Graph Coarsening (MLGC), K-Means (KMEANS).

Datasets	**NCI1**	**NCI109**	**PROTEINS**	**DD**
GIN-SGC	0.23 ± 0.08	0.20 ± 0.05	0.36 ± 0.11	0.25 ± 0.09
GIN-MLGC	0.22 ± 0.07	0.19 ± 0.06	0.36 ± 0.10	0.25 ± 0.10
GIN-SC	0.08 ± 0.11	0.07 ± 0.08	0.32 ± 0.12	0.21 ± 0.10
GIN-KMEANS	0.21 ± 0.09	0.20 ± 0.06	0.35 ± 0.08	0.28 ± 0.09

hyperparameter tuning. It is possible that dataset-specific and model-specific tuning can lead to higher results.

We first obtained the results for Table 1, Figure 2, Table 2, and Table 3, in the main paper. Then, only *afterwards*, we performed the ablation study. The ablation is used to understand the impact of the components of our method only after having evaluated it, as is the standard procedure for ablation studies.

D Full Ablation Study Results

We report the results of the ablation study also for the GIN and GCN models.

Changing Size of Coarsened Graphs. We report in Table 2 and Table 3 the performance of a GCN [6] and GIN [12] model when trained with our regularization strategy with different coarsening ratios C_r. All other hyperparameters are kept the same as for the main results in our paper. As for the PNA model shown in the main paper, we notice an overall trend in which the performance decreases as the coarsening ratio decreases. This follows intuitively as very low coarsening ratios may lead to uninformative graphs. Furthermore we notice that using all ratios (from 0.1 to 0.9) is usually not effective and setting $C = \{0.8, 0.9\}$ seems a good default option.

Changing Coarsening Method. In Table 4 and Table 5 we report the performance of a GCN [6] and a GIN [12] model trained with our regularization using different coarsening functions. All other parameters are kept the same. As for the PNA shown in the main paper we notice that our method is quite robust to the choice of the coarsening function. However, in most cases, specialised methods like SGC and MLGC provide the best results.

Table 6: Average (over ten runs) time (s) for performing one epoch during training of standard GNN models trained with (✓) and without (✗) our regularization method. Standard deviation is not reported as it is small and similar for all results.

Dataset	**NCI1**		**NCI109**		**PROTEINS**		**DD**	
Reg.	✗	✓	✗	✓	✗	✓	✗	✓
PNA	2.02	3.1	3.21	4.94	2.33	3.56	3.45	4.88
GCN	1.92	2.91	2.14	3.88	2.29	3.41	3.26	4.74
GIN	1.95	3.02	2.15	3.86	2.3	3.42	3.41	4.75

E Training Times

Table 6 shows the time in seconds for training a model with and without our regularization. We notice that our method introduces an average 50% overhead in training time.

F Compute and Licence Information

All our experiments were performed on a machine with a Nvidia 1080Ti GPU and a CPU cluster equipped with 8 CPUs 12-core Intel Xeon Gold 5118@2.30GHz with 1.5Tb of RAM.

The TUDataset [9] we used for our experiments are publicly available online[1]. Our code makes use of the following publicly available libraries: PyTorch [10] (BSD-Style License), PyTorch Geometric [4] (MIT License), PyTorch Lightning [3] (Apache-2.0 license), scikit-learn [11] (BSD-3 license), numpy [5] (BSD-3 license), ray[tune] [8] (Apache 2.0).

References

[1] B. Bevilacqua, Y. Zhou, and B. Ribeiro. Size-invariant graph representations for graph classification extrapolations. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 837–851. PMLR, 2021. URL `http://proceedings.mlr.press/v139/bevilacqua21a.html`.

[2] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković. Principal neighbourhood aggregation for graph nets. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13260–13271. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/99cad265a1768cc2dd013f0e740300ae-Paper.pdf`.

[3] W. Falcon et al. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

[4] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric, 2019. URL `https://arxiv.org/abs/1903.02428`.

[5] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[6] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[7] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[1] `https://chrsmrrs.github.io/datasets/`

[8] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

[9] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL `www.graphlearning.io`.

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[12] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

[13] G. Yehudai, E. Fetaya, E. Meirom, G. Chechik, and H. Maron. From local structures to size generalization in graph neural networks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11975–11986. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/yehudai21a.html`.