Data-Driven Self-Supervised Graph Representation Learning

Ahmed E. Samy[a;*], **Zekarias T. Kefato**[a] **and Šarūnas Girdzijauskas**[a]

[a]KTH, Royal Institute of Technology, Stockholm, Sweden
ORCiD ID: Ahmed E. Samy https://orcid.org/0000-0002-5392-6531, Zekarias T. Kefato https://orcid.org/0000-0001-7898-0879, Šarūnas Girdzijauskas https://orcid.org/0000-0003-4516-7317

A Training Equations

$$\texttt{inv} = ||\mathbf{Z}_1 - \mathbf{Z}_2||_F \tag{1}$$

$$v(\mathbf{Z}) = \frac{1}{D}\sum_{j=1}^{D}\max(0, 1 - \sqrt{Var(\mathbf{z}_{:j}) + \epsilon}) \tag{2}$$

$$c(\mathbf{Z}) = \frac{1}{D}\sum_{i\neq j}\Big[\frac{\bar{\mathbf{Z}}^T\bar{\mathbf{Z}}}{B-1}\Big]_{i,j}^2 \tag{3}$$

$$R_{\mathbf{Z}_1,\mathbf{Z}_2} = \beta * (v(\mathbf{Z}_1) + v(\mathbf{Z}_2)) + \gamma * (c(\mathbf{Z}_1) + c(\mathbf{Z}_2)) \tag{4}$$

$$R_{\Theta_1,\Theta_2} = \sum_{\mathbf{W}_l}||\mathbf{W}_l\mathbf{W}_l^T - \mathbf{I}||_F \tag{5}$$

$$\mathcal{L}_\Psi = \alpha * \texttt{inv} + R_{\mathbf{Z}_1,\mathbf{Z}_2} + \lambda * R_{\Theta_1,\Theta_2} \tag{6}$$

$$R_{\mathbf{Z}_1,\mathbf{Z}_2} = \gamma * (||\tilde{\mathbf{Z}}_\mathbf{1}\tilde{\mathbf{Z}}_\mathbf{1}^T - \mathbf{I}||_F + ||\tilde{\mathbf{Z}}_\mathbf{2}\tilde{\mathbf{Z}}_\mathbf{2}^T - \mathbf{I}||_F) \tag{7}$$

B Appendix

B.0.1 Loss Function Coefficients

Because of the strong inductive bias from the GNN encoder, h_Θ, in some cases, the loss function terms are not very sensitive to the coefficients α, β, γ. Our empirical analysis on Table 1 shows that having one of the regularizers could be sufficient to avoid collapse. This is corroborated by the fact that the performance of the untrained models, Random-F and Random-T, is strong. For the NC task, both on small and large datasets, we observe that turning off the regularizers often leads to collapse, as shown in Table 2. Mainly, γ plays a crucial role in achieving this.

In the case of λ, although it was introduced for the theoretical completeness of the model, empirically, we only observe a negligible impact on all the datasets. Furthermore, we did not observe major difference in using Eq. 6 and 7, as long as the proper model selection is carried out.

* Corresponding Author. Email: aesy@kth.se

Table 1. Loss function (Eq. 6) coefficients analysis.

Datasets	α	β	γ	Accuracy
ENZYMES	1	1	1	54.8±3.7
	1	1	0	52.6±5.5
	1	0	1	54.1±3.8
	1	0	0	54.8±5.6
	0	0	1	54.8±3.3
	0	1	0	46.1±4.4
DD	1	1	1	78.3±3.8
	1	1	0	78.2±4.2
	1	0	1	78.2±3.7
	1	0	0	78.2±3.6
	0	0	1	78.2 ±3.9
	0	1	0	77.7±3.9
IMDB-M	1	1	1	50.8±3.6
	1	1	0	49.1±4.2
	1	0	1	50.2±3.2
	1	0	0	49.6±3.5
	0	0	1	50.0±3.5
	0	1	0	47.9±4.6

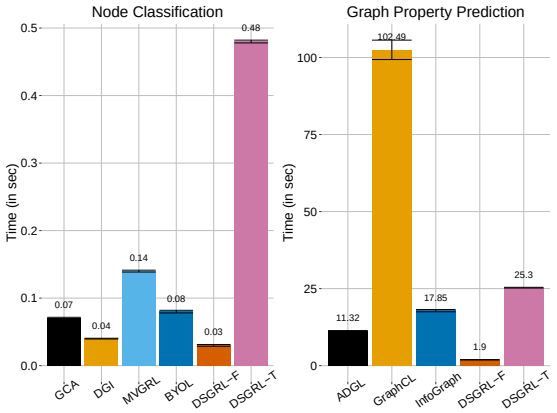
Table 2. Loss function (Eq. 7) coefficients analysis.

Dataset	α	γ	Accuracy
CS	1	1	91.1±0.06
	1	0	88.5±0.1
	0	1	91.1±0.06
PubMed	1	1	81.2±0.2
	1	0	70.6±0.6
	0	1	80.5±0.002
Reddit	1	1	89.3
	1	0	59.6
	0	1	82.4

Table 3. Batch size analysis

Datasets	Batch Size	Accuracy
ENZYMES	16	56.5±6.7
	32	54.0±3.3
	64	50.8±5.3
	128	53.5±5.0
	256	52.3±4.3
DD	16	75.5±3.9
	32	78.2±3.7
	64	77.1±4.0
	128	77.2±3.5
	256	76.9±3.6
IMDB-M	16	48.0±4.4
	32	50.3±3.4
	64	50.4±3.5
	128	50.6±3.6
	256	49.8±3.6

Figure 1. The average run-time to finish an epoch for the ssl methods



B.0.2 Batch Size

One of the limitations of contrastive methods based on negative sampling is that they usually require large batch-size. We study the effect of different values to demonstrate that our approach does not require a large batch size. As shown in Table 3, having a large batch size has no benefit. In fact, We have observed that large batch sizes require more epochs to converge.

B.1 Ablation Studies

Finally, we examine the run-time of DSGRL by comparing it against the self-supervised baselines. For this experiment, we consider the average run time of 10 trials that an algorithm requires to finish a single training epoch.

The results for both NC and GPP on the AmazonPhoto and DD datasets, respectively, are reported in Fig. 1. As shown in the figure, DSGRL-F, which employs a learnable feature augmentation technique, is faster than all other methods in both experiments. In particular, it has more than an order of magnitude gain for the graph classification experiment. However, since the learnable topology augmentation requires a quadratic complexity just for learning the augmentation, DSGRL-T is not scalable for large graphs. Nonetheless, considering that (1) both augmentation techniques are qualitatively comparable and (2) topology based augmentations are susceptible to violating the inherent property of molecular graphs; one can safely rely on feature-based augmentations, which have no trade-offs.

The values of the hyper-parameters of DSGRL used in our empirical evaluation are shown in Tables 4, 5, 6, and 7. Float values are rounded up for presentation; the exact values are provided in the accompanying source code. Furthermore, the source code includes a model selection component. Thus, one can also tune DSGRL as necessary.

D_1, D, and L_h denote the augmenter's output dimension, the model's final output dimension, and the number of GNN layers, respectively. Note that, Tables 4 and 5 are based on Eq. 6 and Tables 6 and 7 are based on Eq. 7.

We did not rely on complicated and advanced models for a fair comparison. Therefore, for the node classification experiment, we use the vanilla GCN [**?**] on the small datasets and GRAPHSAGE [**?**] for large ones. For graph property prediction we use GIN [**?**] and global add pooling to obtain a single representation for each graph.

Table 4. The hyper-parameter configuration of DSGRL with feature augmentation for graph property prediction

Dataset	D_1	D	L_h	α	β	γ	λ	Dropout	Learning Rate	Early Termination Epoch	Evaluation Epochs
DD	90	53	1	1	1	1	1	0.6	0.0002	1	100
NCI1	128	58	2	18	57	58	11	0.9	0.008	5	100
PROTEINS	450	34	3	43	77	16	16	0.8	0.0005	1	100
ENZYMES	93	64	1	55	30	61	34	0.2	0.002	5	100
IMDB-B	128	64	2	44	34	20	65	0.07	0.01	1	100
IMDB-M	459	46	3	18	67	100	94	0.7	0.01	10	100
REDDIT	128	64	3	37	27	61	33	0.4	0.007	2	100
COLLAB	151	51	2	28	92	70	81	0.8	0.01	2	100

Table 5. The hyper-parameter configuration of DSGRL with topology augmentation for graph property prediction

Dataset	D_1	D	L_h	α	β	γ	λ	Dropout	Learning Rate	Early Termination Epoch	Evaluation Epochs
DD	128	64	4	1	1	1	1	0.4	0.001	4	100
NCI1	402	60	2	32	28	57	26	0.1	0.003	1	100
PROTEINS	241	45	1	43	77	16	16	0.9	0.01	20	100
ENZYMES	93	64	1	55	30	61	34	0.2	0.002	10	100
IMDB-B	128	64	3	8	68	6	73	0.4	0.01	1	100
IMDB-M	128	64	3	21	63	21	73	0.4	0.05	11	100
REDDIT	128	64	3	37	27	61	33	0.4	0.007	3	100
COLLAB	151	51	2	28	92	70	81	0.8	0.01	1	100

Table 6. The hyper-parameter configuration of DSGRL with feature augmentation for node classification

Dataset	D_1	D	L_h	α	γ	λ	Dropout	Learning Rate	Training Epochs	Evaluation Epochs
CS	256	64	2	1	1	1	0.5	0.001	100	100
Photo	256	64	2	1	1	1	0.5	0.001	100	100
PubMed	445	64	2	0.6	0.5	1	0.3	0.006	100	100
GitHub	360	64	2	0.3	0.8	1	0.8	0.06	100	100
WikiCS	128	64	2	1	1	1	0.5	0.001	100	100
Deezer	92	64	2	0.9	0.5	1	0.4	0.05	100	100
Yelp	128	64	2	1	1	1	0.5	0.001	5	500
Reddit	128	64	2	1	1	1	0.5	0.001	1	500

Table 7. The hyper-parameter configuration of DSGRL with topology augmentation for node classification

Dataset	D_1	D	L_h	α	γ	λ	Dropout	Learning Rate	Training Epochs	Evaluation Epochs
CS	327	64	2	0.8	0.01	0.4	0.1	0.006	100	100
Photo	260	64	2	0.3	0.1	0.6	0.1	0.004	100	100
PubMed	387	64	2	0.7	0.2	0.5	0.2	0.02	100	100
GitHub	107	64	2	0.9	0.8	0.4	0.9	0.06	100	100
WikiCS	451	64	2	0.9	0.1	0.05	0.1	0.01	100	100
Deezer	417	64	2	0.6	0.9	0.5	0.6	0.02	100	100