

APPENDIX

This appendix provides additional information not described in the main text due to the page limit. It contains proofs in Section A, various experimental details in Section B, more quantitative results on benchmarks in Section C and more qualitative results of image retrieval with visual scene graphs in Section D.

A PROOF

A.1 PROOF FOR PROPOSITION 1

Proof. Suppose we have a coarsening matrix $P \in \{0, 1\}^{N \times n}$. Similar to Loukas (2019), to show that matrix $\Pi = PQ^{-2}P^\top$ is a projection matrix, i.e., $\Pi^2 = \Pi$, of rank n is a sufficient condition to prove $P^+ = Q^{-2}P^\top$, where P has rank n . Suppose we particularly sort rows of P , such that for any two columns $r < r'$, if $P(i, r) = 1$ then $P(i', r') = 0$ for $i' < i$. Furthermore, denoted by p_r the 1 vector containing all 1 entries of $P(:, r)$ such that $\|p_r\|_2 = \|P(:, r)\|_2 = Q(r, r)$. Then the $B_r = p_r Q(r, r)^{-2} p_r^\top$ is a rank 1 projection matrix as $B_r^2 = B_r B_r = (p_r Q(r, r)^{-2} p_r^\top) (p_r Q(r, r)^{-2} p_r^\top) = p_r Q(r, r)^{-2} p_r^\top \frac{p_r^\top p_r}{\|p_r\|_2^2} = B_r$. Finally, the matrix Π is a block-diagonal matrix (in our particular sorting), $\Pi = \text{diag}(B_1, \dots, B_n)$, so that the $\Pi^2 = \Pi$ is a projection matrix of rank n .

A.2 PROOF FOR EQUATION 13

proof. referred to Loukas (2019), by the triangle inequality:

$$\begin{aligned} ||u||_L - ||u_c||_{L_c} &= \left| \sqrt{u^\top L u} - \sqrt{u_c^\top L_c u_c} \right| = \left| \sqrt{u^\top L u} - \sqrt{u^\top P^\top P^\top L P P^\top u} \right| \\ &= \left| \sqrt{u^\top L u} - \sqrt{\tilde{u}^\top L \tilde{u}} \right| \leq \left| \sqrt{(u - \tilde{u})^\top L (u - \tilde{u})} \right| = ||(u - \tilde{u})||_L \end{aligned} \quad (14)$$

B EXPERIMENTAL DETAILS

B.1 GRAPH POOLING BASELINE MODELS

For the experiments, we use five competitive baselines recently proposed for differentiable graph pooling: SortPool (Zhang et al., 2018), gPool (Gao & Ji, 2019), SAGPool (Lee et al., 2019), EdgePool (Diehl, 2019) and DiffPool (Ying et al., 2018). The details of each baselines are shown below:

SortPool (Zhang et al., 2018): SortPool is a spatial pooling method that sorts nodes in spatial order according to their structural roles. After sorting, they rearrange truncated k nodes from the graph and feed it to a following 1D convolutional layer.

gPool (Gao & Ji, 2019): gPool is a top- k pooling operation that samples a set of important nodes from a graph. gPool samples k nodes according to the score based on a trainable projection vector. In this experiment, the parameter k was selected from the pooling ratios of other models.

SAGPool (Lee et al., 2019): A variant of gPool, SAGPool considers both node feature and graph topology for sampling nodes. SAGPool computes self attention scores between nodes after passing through graph convolution layers to select top- k important nodes.

EdgePool (Diehl et al., 2019): EdgePool is a pooling method that utilizes edge contraction. Edge contraction is to choose an edge based on a certain edge scoring function and combine two nodes that are linked with it. Unlike other models, the pooling ratio is fixed.

DiffPool (Ying et al., 2018): DiffPool is an end-to-end differentiable pooling method that learns hierarchical representations of graphs. DiffPool makes use of a trainable soft assignment matrix that regards both node feature and graph topology.

B.2 IMPLEMENTATION DETAILS

For the graph classification task, we evaluated all the comparative models over 10 random seeds using 10-fold cross validation. The total of 100 testing results was used to obtain the final accuracy of each method on each dataset. 10 percent of the training data was used for validation in the training session. In order to ensure a fair comparison, both the proposed and comparative pooling methods are implemented through the same neural network architecture with same GNN module. We use Adam optimizer with $1e - 3$ learning rate, and the models are trained for 100 epochs with batch size 16. To get optimal hyper-parameters for each model, the number of pooling layer $K \in \{1, 2\}$, pooling ratio $r \in \{0.1, 0.2, 0.4, 0.8\}$ and 32 dimensional feature vectors nodes are used. A 1-pooling layer architecture is composed of {Graph Convolution - Graph Pooling - Graph Convolution} and 2-pooling layer architecture is composed of {Graph Convolution - Graph Pooling - Graph Convolution - Graph Pooling - Graph Convolution} with GNN modules.

For the image retrieval task, we compared our model with two representative pooling algorithms, DiffPool and SAGPool. To get optimal hyper-parameters for each model, A single pooling layer is used with pooling ratio $r \in \{0.1, 0.2, 0.4, 0.8\}$ and 300 dimensional feature vectors nodes are used. We use Adam optimizer with $1e - 4$ learning rate, and the models are trained for 30 epochs with batch size 32.

B.3 STATISTICS OF GRAPH BENCHMARK DATASETS

The statistics of graph benchmark datasets used in experiments are shown in Table 3. As we noted in Section 5.1, we chose four datasets according to their amount of data and graph size to evaluate the generality of the proposed model.

Table 3: Statistics of Graph Benchmark Datasets

Datasets	# of graphs	# of classes	# of avg. nodes	# of avg. edges
MUTAG	188	2	17.93	19.79
ENZYMES	600	6	32.63	62.14
PROTEINS	1,113	2	39.06	72.82
NCI1	4,110	2	29.87	32.30

C RESULTS FOR BENCHMARKS WITH VARYING HYPERPARAMETERS

Table 4 shows the comparison results along with pooling ratio , the number of pooling layers and the existence of regularizer for each dataset. we observed that the proposed regularization term usually improves performance most of configurations across the all datasets. This implies that preserving graph topology while simultaneously pooling graphs has a substantial impact on graph representation learning.

D QUALITATIVE RESULTS OF IMAGE RETRIEVAL TASK

Here, we present additional qualitative results of graph pooling for the model compared to DiffPool and SAGPool in Figure 4. In Figure 4, we omit node labels to compare the overall pooling characteristics in each model. As shown, the SSGPool can coarsen the graph by preserving the global structure of graphs, but DiffPool and SAGPool can not. It is worthwhile to notice that the coarsened graphs of Diffpool are always fully-connected and those of SAGPool also hardly capture the global structure, thus the coarsened graph could be disconnected even though the original one is connected.

Table 4: Effect of proposed regularizer in SSGPool for various benchmark datasets along with pooling ratio and the number of pooling layer.

		Ratio (0.1)	Ratio (0.2)	Ratio (0.4)	Ratio (0.8)
Pooling 1	w/o Reg.	0.844±0.019	0.834±0.015	0.839±0.019	0.846 ±0.015
	w/ Reg.	0.847 ±0.014	0.845 ±0.013	0.844 ±0.014	0.844±0.012
Pooling 2	w/o Reg.	0.841±0.017	0.845±0.025	0.841±0.027	0.825±0.032
	w/ Reg.	0.844 ±0.011	0.852 ±0.009	0.846 ±0.008	0.850 ±0.010

(a) Classification results for MUTAG dataset.

		Ratio (0.1)	Ratio (0.2)	Ratio (0.4)	Ratio (0.8)
Pooling 1	w/o Reg.	0.362±0.037	0.364±0.002	0.348±0.040	0.369±0.021
	w/ Reg.	0.382 ±0.012	0.378 ±0.022	0.377 ±0.034	0.375 ±0.028
Pooling 2	w/o Reg.	0.348±0.023	0.352±0.020	0.302±0.052	0.317 ±0.033
	w/ Reg.	0.363 ±0.025	0.374 ±0.029	0.343 ±0.016	0.303±0.025

(b) Classification results for ENZYMES dataset.

		Ratio (0.1)	Ratio (0.2)	Ratio (0.4)	Ratio (0.8)
Pooling 1	w/o Reg.	0.735±0.010	0.741 ±0.008	0.728±0.014	0.723±0.012
	w/ Reg.	0.750 ±0.010	0.736±0.013	0.741 ±0.009	0.735 ±0.014
Pooling 2	w/o Reg.	0.733±0.011	0.745 ±0.006	0.742±0.007	0.742±0.006
	w/ Reg.	0.743 ±0.009	0.744±0.003	0.750 ±0.007	0.750 ±0.005

(c) Classification results for PROTEINS dataset.

		Ratio (0.1)	Ratio (0.2)	Ratio (0.4)	Ratio (0.8)
Pooling 1	w/o Reg.	0.738 ±0.019	0.738 ±0.018	0.734 ±0.015	0.729 ±0.018
	w/ Reg.	0.735±0.022	0.735±0.012	0.731±0.014	0.728±0.017
Pooling 2	w/o Reg.	0.742±0.014	0.703±0.041	0.737±0.015	0.752±0.005
	w/ Reg.	0.746 ±0.010	0.743 ±0.007	0.753 ±0.010	0.753 ±0.012

(d) Classification results for NCI1 dataset.

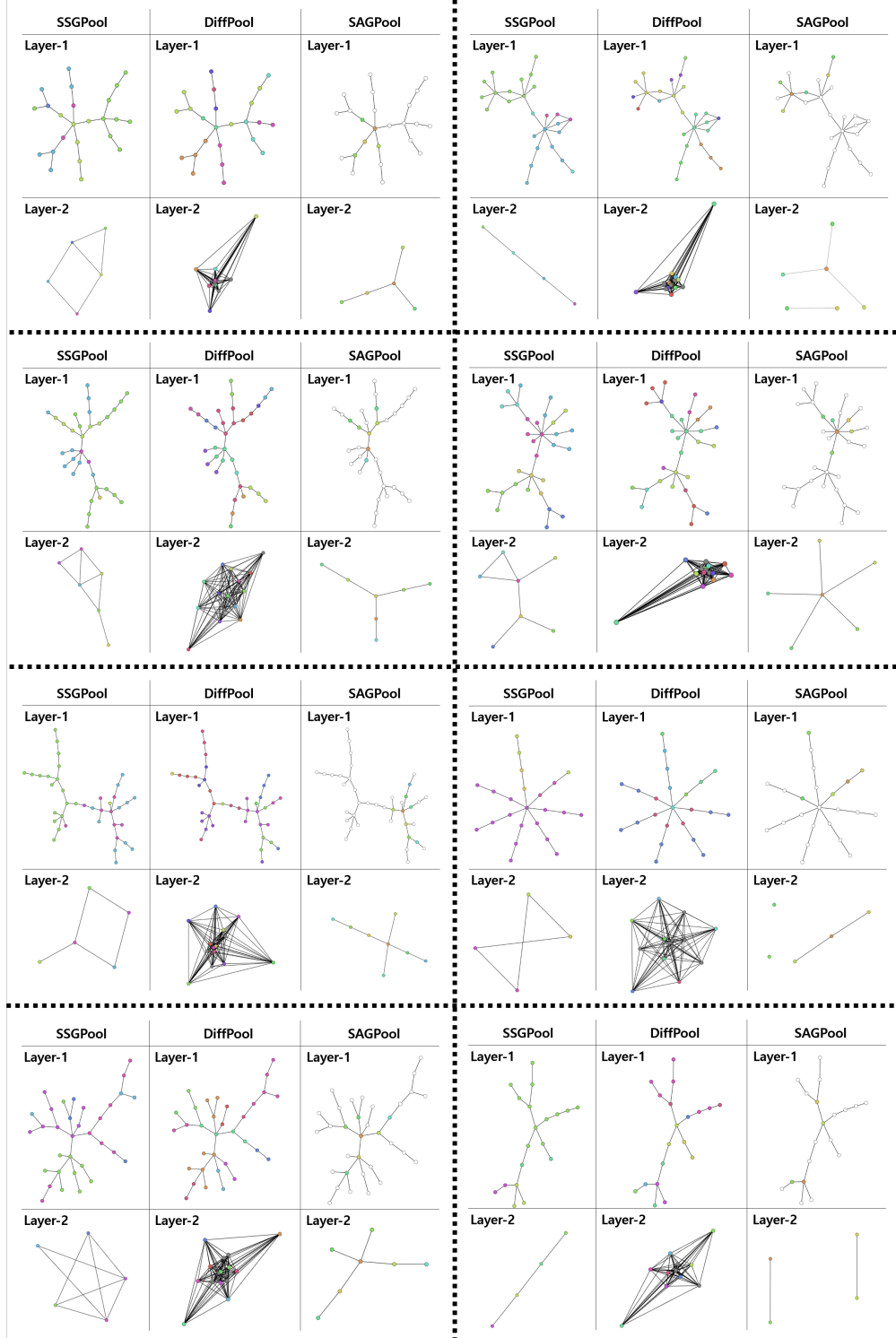


Figure 4: Other qualitative examples of pooling results on SSGPool, DiffPool and SAGPool. As Figure 3, same color of nodes are meant to be mapped to the same coarsened node in the pooled layer.