Group Meeting - November 13, 2020

Paper review & Research progress

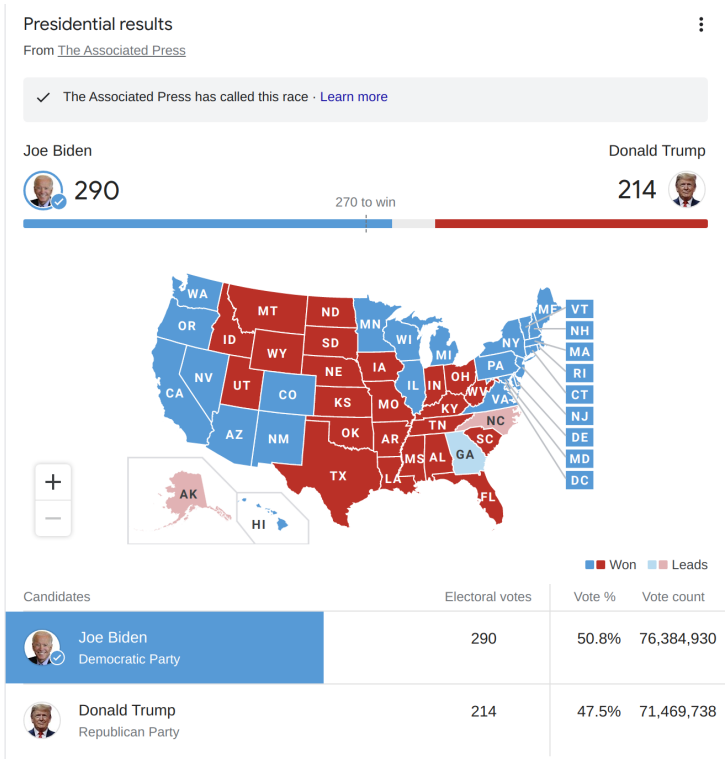
Truong Son Hy *

*Department of Computer Science
The University of Chicago

Ryerson Physical Lab

Buddha

1. If you want to be successful, you must respect one rule: **Never lie to yourself.**
2. All human **unhappiness** comes from **not facing reality** squarely, exactly as it is.
3. Ego never accepts the truth.

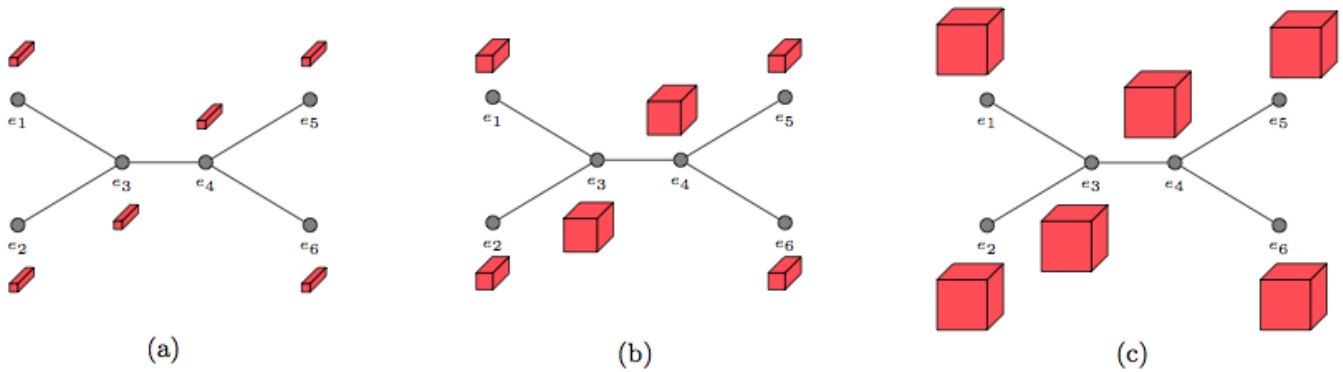


Papers

1. **Self-Supervised Graph Transformer on Large-Scale Molecular Data** (NeurIPS 2020)
   GROVER: Self-supervised Message Passing Transformer on Large-scale Molecular Data (preprint)
   `https://arxiv.org/abs/2007.02835`
2. **A Flexible Generative Framework for Graph-based Semi-supervised Learning** (NeurIPS 2019),
   `https://arxiv.org/abs/1905.10769`

Brainstorm (1)

Problem: How input vertex/atom featurization affects the adjacency reconstruction. Let consider the Ethylene (C_2H_4) molecular graph for example example.



- The input atom features of 4 hydrogen atoms are **identical** (the same). The input atom features of 2 carbon atoms are also the same.
- Because of the symmetry, all the tensors (let's consider CCNs for example) associating with hydrogen/carbon atoms are the same.

Brainstorm (2)

Problem

There is an edge between e1 (hydrogen) and e3 (carbon). But there is no edge between e5 (hydrogen) and e3 (carbon). Therefore, there is **NO** function can classify edge/non-edge for pairs (e1, e3) and (e5, e3).

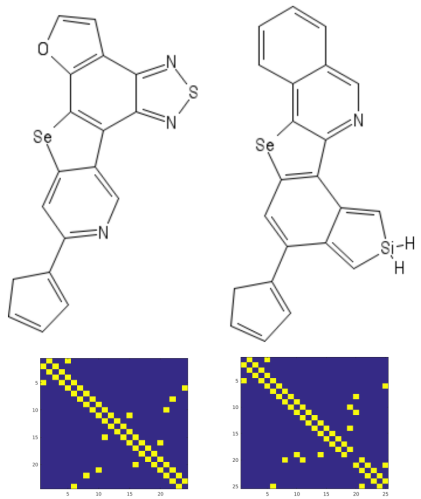
Experiment

- Use Support Vector Machine (SVM) or simple logistic regression
- To classify edge/non-edge.
- The data are pairs of input atomic features (2 atoms).
- We want to know what is the minimum amount of atomic features needed for each atom to accurately predict edge/non-edge.
- SVM would give us a **reasonable baseline** for the auto-encoding task.
- **Dictionary Weisfeiler-Lehman** to enrich the input features.

Brainstorm (3)

Markov Random Field (MRF):

- 3/5 have aromatic rings. 2/5 have Benzene. I am thinking of **why**?
- Moving to **Harvard Clean Energy Project** (HCEP) dataset that has a lot of rings?



Paper 1

Self-Supervised Graph Transformer on Large-Scale Molecular Data

(NeurIPS 2020)

Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, Junzhou Huang

`https://arxiv.org/abs/2007.02835`

Proposals

Problem of GNNs

1. Insufficient labeled molecules for supervised training.
2. Poor generalization capabilities to new-synthesized molecules.

Proposals

GROVER = **G**raph **r**epresentation fr**o**m self-supser**v**ised m**e**ssage passing t**r**ansformer.

- Self-supervised tasks in node, edge and graph-level.
- Transformer-style architecture.
- Pre-train GROVER with 100M parameters on 10M unlabelled molecules $\rightarrow$ the biggest GNN and the largest training dataset.

Multi-head Attention

The multi-head attention mechanism is the main building block of various Transformer - style models:

- Parallel running.
- Stacks several scaled dot-product attention layers together. One scaled dot-product attention layer takes a set of queries, keys, values $(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v})$ as inputs:

$$\text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d})\boldsymbol{V}$$

- Suppose we arrange k attention layers into the multi-head attention:

$$\text{Multi-Head}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{Concat}(\text{head}_1, .., \text{head}_k)\boldsymbol{W}^O$$

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V)$$

where $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$ and $\boldsymbol{W}_i^V$ are the projection matrices of head i.

Graph Neural Networks (GNNs)

Suppose there are L **iterations**, and iteration ℓ contains K_ℓ **hops**:

$$\boldsymbol{m}_v^{(\ell,k)} = \text{AGGREGATE}(\{(\boldsymbol{h}_v^{(\ell,k-1)}, \boldsymbol{h}_u^{(\ell,k-1)}, \boldsymbol{e}_{uv}) | u \in \mathcal{N}_v\})$$

$$\boldsymbol{h}_v^{(\ell,k)} = \sigma(\boldsymbol{W}^{(\ell)} \boldsymbol{m}_v^{(\ell,k)} + \boldsymbol{b}^{(\ell)})$$

READOUT operation is applied to get the graph - level representation:

$$\boldsymbol{h}_G = \text{READOUT}(\{\boldsymbol{h}_v^{(0,K_0)}, .., \boldsymbol{h}_v^{(L,K_L)} | v \in \mathcal{V}\})$$

Dynamic Message Passing Network (dyMPN):

- The number of hops is closely related to the size of the receptor field.
- Instead of pre-specifying the number of hops, we develop a **randomized strategy** for choosing the number of hops.

GNN Transformer (GTransformer)

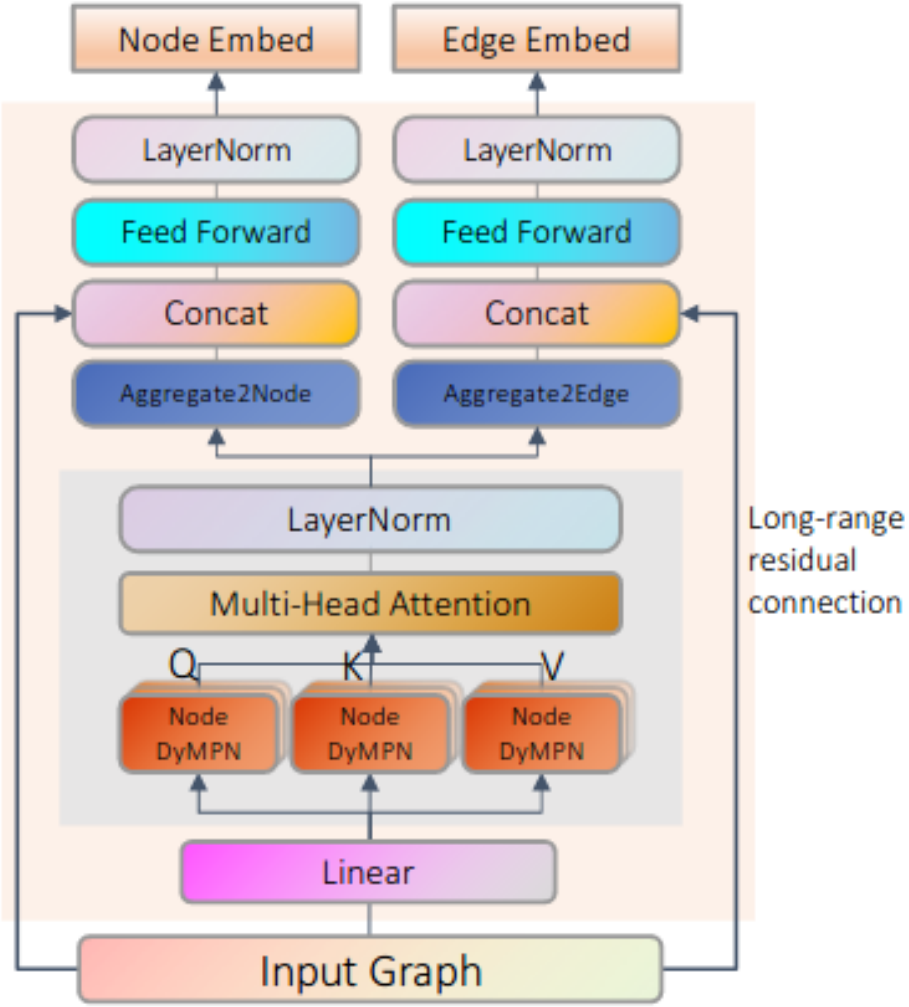


Figure 1: Overview of GTransformer.

Self-supervised Tasks

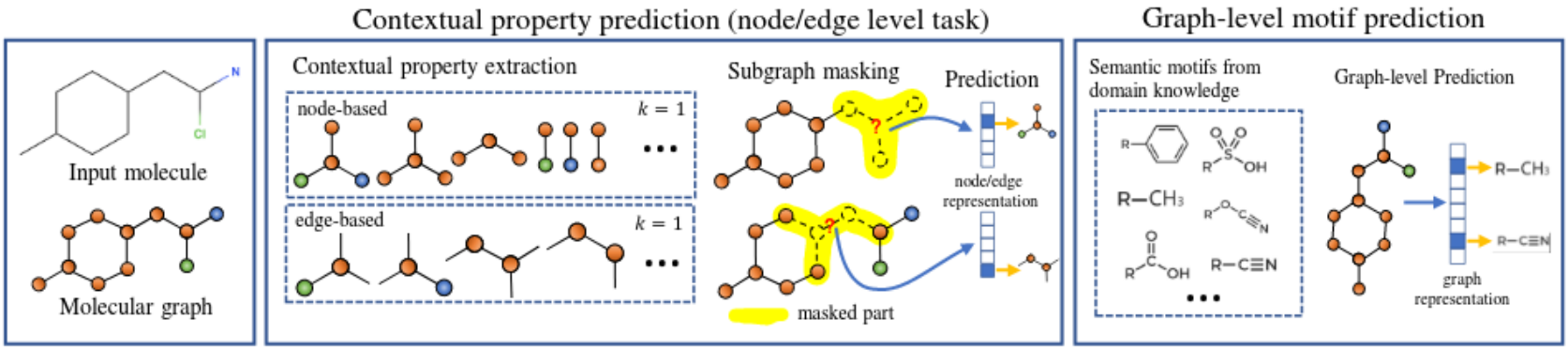


Figure 2: Overview of the designed self-supervised tasks of GROVER.

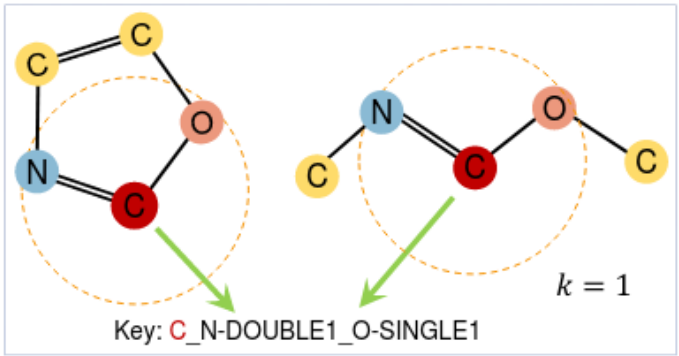


Figure 3: Illustration of contextual properties.

Benchmark datasets

http://moleculenet.ai/datasets-1

Dataset Details

Category	Dataset	Data Type	Task Type	# Tasks	# Compounds	Rec - Split[a]	Rec - Metric[b]
Quantum Mechanics	QM7	SMILES, 3D coordinates	Regression	1	7160	Stratified	MAE
	QM7b	3D coordinates	Regression	14	7210	Random	MAE
	QM8	SMILES, 3D coordinates	Regression	12	21786	Random	MAE
	QM9	SMILES, 3D coordinates	Regression	12	133885	Random	MAE
Physical Chemistry	ESOL	SMILES	Regression	1	1128	Random	RMSE
	FreeSolv	SMILES	Regression	1	642	Random	RMSE
	Lipophilicity	SMILES	Regression	1	4200	Random	RMSE
Biophysics	PCBA	SMILES	Classification	128	437929	Random	PRC-AUC
	MUV	SMILES	Classification	17	93087	Random	PRC-AUC
	HIV	SMILES	Classification	1	41127	Scaffold	ROC-AUC
	PDBbind	SMILES, 3D coordinates	Regression	1	11908	Time	RMSE
	BACE	SMILES	Classification	1	1513	Scaffold	ROC-AUC
Physiology	BBBP	SMILES	Classification	1	2039	Scaffold	ROC-AUC
	Tox21	SMILES	Classification	12	7831	Random	ROC-AUC
	ToxCast	SMILES	Classification	617	8575	Random	ROC-AUC
	SIDER	SMILES	Classification	27	1427	Random	ROC-AUC
	ClinTox	SMILES	Classification	2	1478	Random	ROC-AUC

Pre-training performance

Table 1: The performance comparison. The numbers in brackets are the standard deviation. The methods in green are pre-trained methods.

Classification (Higher is better)						
Dataset	BBBP	SIDER	ClinTox	BACE	Tox21	ToxCast
# Molecules	2039	1427	1478	1513	7831	8575
TF_Robust [40]	$0.860_{(0.087)}$	$0.607_{(0.033)}$	$0.765_{(0.085)}$	$0.824_{(0.022)}$	$0.698_{(0.012)}$	$0.585_{(0.031)}$
GraphConv [24]	$0.877_{(0.036)}$	$0.593_{(0.035)}$	$0.845_{(0.051)}$	$0.854_{(0.011)}$	$0.772_{(0.041)}$	$0.650_{(0.025)}$
Weave [23]	$0.837_{(0.065)}$	$0.543_{(0.034)}$	$0.823_{(0.023)}$	$0.791_{(0.008)}$	$0.741_{(0.044)}$	$0.678_{(0.024)}$
SchNet [45]	$0.847_{(0.024)}$	$0.545_{(0.038)}$	$0.717_{(0.042)}$	$0.750_{(0.033)}$	$0.767_{(0.025)}$	$0.679_{(0.021)}$
MPNN [13]	$0.913_{(0.041)}$	$0.595_{(0.030)}$	$0.879_{(0.054)}$	$0.815_{(0.044)}$	$0.808_{(0.024)}$	$0.691_{(0.013)}$
DMPNN [63]	$0.919_{(0.030)}$	$0.632_{(0.023)}$	$0.897_{(0.040)}$	$0.852_{(0.053)}$	$0.826_{(0.023)}$	$0.718_{(0.011)}$
MGCN [30]	$0.850_{(0.064)}$	$0.552_{(0.018)}$	$0.634_{(0.042)}$	$0.734_{(0.030)}$	$0.707_{(0.016)}$	$0.663_{(0.009)}$
AttentiveFP [61]	$0.908_{(0.050)}$	$0.605_{(0.060)}$	$0.933_{(0.020)}$	$0.863_{(0.015)}$	$0.807_{(0.020)}$	$0.579_{(0.001)}$
N-GRAM [29]	$0.912_{(0.013)}$	$0.632_{(0.005)}$	$0.855_{(0.037)}$	$0.876_{(0.035)}$	$0.769_{(0.027)}$	-[4]
HU. et.al[18]	$0.915_{(0.040)}$	$0.614_{(0.006)}$	$0.762_{(0.058)}$	$0.851_{(0.027)}$	$0.811_{(0.015)}$	$0.714_{(0.019)}$
$\text{GROVER}_{\text{base}}$	$0.936_{(0.008)}$	$0.656_{(0.006)}$	$0.925_{(0.013)}$	$0.878_{(0.016)}$	$0.819_{(0.020)}$	$0.723_{(0.010)}$
$\text{GROVER}_{\text{large}}$	$\mathbf{0.940}_{(0.019)}$	$\mathbf{0.658}_{(0.023)}$	$\mathbf{0.944}_{(0.021)}$	$\mathbf{0.894}_{(0.028)}$	$\mathbf{0.831}_{(0.025)}$	$\mathbf{0.737}_{(0.010)}$

Regression (Lower is better)					
Dataset	FreeSolv	ESOL	Lipo	QM7	QM8
# Molecules	642	1128	4200	6830	21786
TF_Robust [40]	$4.122_{(0.085)}$	$1.722_{(0.038)}$	$0.909_{(0.060)}$	$120.6_{(9.6)}$	$0.024_{(0.001)}$
GraphConv [24]	$2.900_{(0.135)}$	$1.068_{(0.050)}$	$0.712_{(0.049)}$	$118.9_{(20.2)}$	$0.021_{(0.001)}$
Weave [23]	$2.398_{(0.250)}$	$1.158_{(0.055)}$	$0.813_{(0.042)}$	$94.7_{(2.7)}$	$0.022_{(0.001)}$
SchNet [45]	$3.215_{(0.755)}$	$1.045_{(0.064)}$	$0.909_{(0.098)}$	$74.2_{(6.0)}$	$0.020_{(0.002)}$
MPNN [13]	$2.185_{(0.952)}$	$1.167_{(0.430)}$	$0.672_{(0.051)}$	$113.0_{(17.2)}$	$0.015_{(0.002)}$
DMPNN [63]	$2.177_{(0.914)}$	$0.980_{(0.258)}$	$0.653_{(0.046)}$	$105.8_{(13.2)}$	$0.0143_{(0.002)}$
MGCN [30]	$3.349_{(0.097)}$	$1.266_{(0.147)}$	$1.113_{(0.041)}$	$77.6_{(4.7)}$	$0.022_{(0.002)}$
AttentiveFP [61]	$2.030_{(0.420)}$	$0.853_{(0.060)}$	$0.650_{(0.030)}$	$126.7_{(4.0)}$	$0.0282_{(0.001)}$
N-GRAM [29]	$2.512_{(0.190)}$	$1.100_{(0.160)}$	$0.876_{(0.033)}$	$125.6_{(1.5)}$	$0.0320_{(0.003)}$
$\text{GROVER}_{\text{base}}$	$1.592_{(0.072)}$	$0.888_{(0.116)}$	$0.563_{(0.030)}$	$\mathbf{72.5}_{(5.9)}$	$0.0172_{(0.002)}$
$\text{GROVER}_{\text{large}}$	$\mathbf{1.544}_{(0.397)}$	$\mathbf{0.831}_{(0.120)}$	$\mathbf{0.560}_{(0.035)}$	$72.6_{(3.8)}$	$\mathbf{0.0125}_{(0.002)}$

Comparing to its variants

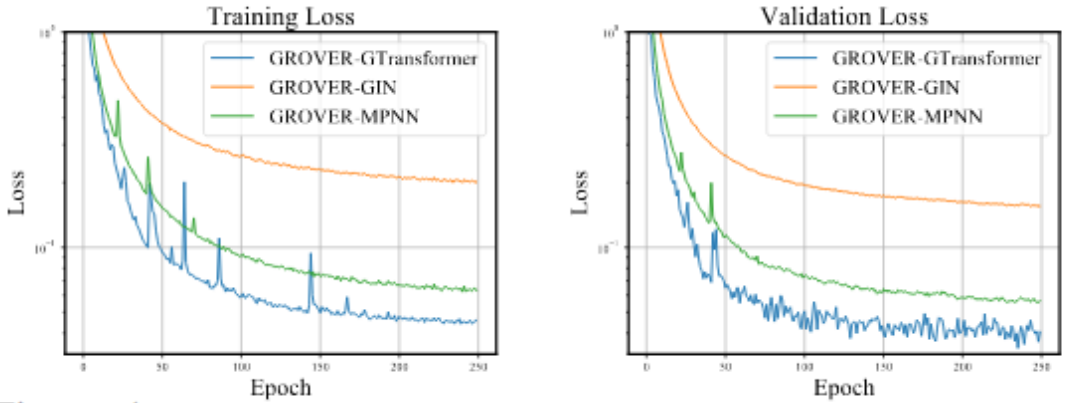


Figure 4: The training and validation losses on different backbones.

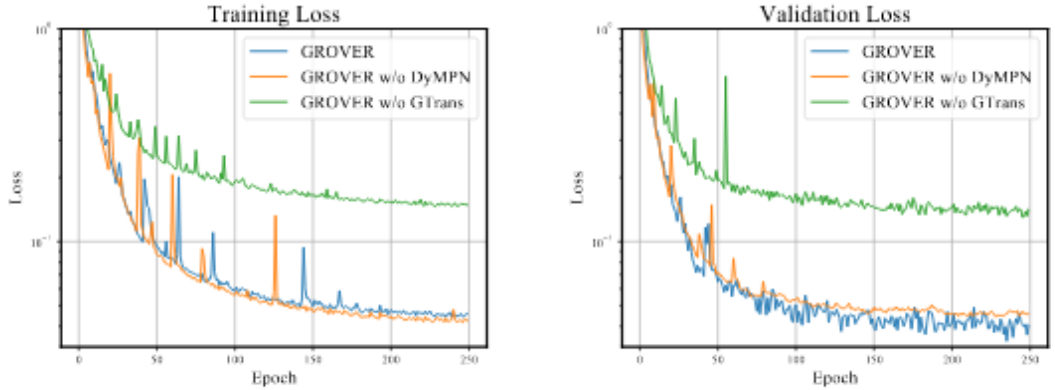


Figure 5: The training and validation loss of GROVER and its variants.

Comparing to without pre-training

	GROVER	No Pretrain	Abs. Imp.
BBBP (2039)	**0.940**	0.911	+0.029
SIDER (1427)	**0.658**	0.624	+0.034
ClinTox (1478)	**0.944**	0.884	+0.060
BACE (1513)	**0.894**	0.858	+0.036
Tox21 (7831)	**0.831**	0.803	+0.028
ToxCast (8575)	**0.737**	0.721	+0.016
Average	**0.834**	0.803	+0.038

Table 2: Comparison between GROVER with and without pre-training.

Paper 2

A Flexible Generative Framework for Graph-based Semi-supervised Learning (NeurIPS 2019)

Jiaqi Ma, Weijing Tang, Ji Zhu, Qiaozhu Mei

`https://arxiv.org/abs/1905.10769`

Proposals

Proposals

1. Generative framework for graph-based SSL.
2. Node features, labels, and graph structure are modeled in a joint distribution.
3. Employ the variational inference techniques to approximate the Bayesian posterior.

Note

The datasets in the experiment are too small. The work should be scalable for a much larger network like IMAGENET.

Graph-based regularization for semi-supervised learning

Suppose there are n data samples in total and m of them are labeled, the graph-based regularization methods generally conduct semi-supervised learning by optimizing the objective:

$$\sum_{i=1}^{m} \mathcal{L}_i + \eta \sum_{i,j=1}^{n} w_{i,j} \mathcal{R}(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

where:

- $\mathcal{L}_i$ is the supervised loss function of sample i.
- $\mathcal{R}(\cdot, \cdot)$ is a regularization function.
- w_{ij} is a graph-based coefficient.
- $\boldsymbol{f}_i, \boldsymbol{f}_j$ are the outcome predictions or feature representations.
- η is a hyper-parameter trade-off the supervised loss and the graph-based regularization.

A flexible generative framework for graph-based SSL (1)

Denote $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $\boldsymbol{Y} \in \mathbb{R}^{n \times \ell}$ as feature and outcome vectors. Partition the outcome matrix as:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{Y}_{obs} \\ \boldsymbol{Y}_{miss} \end{bmatrix}$$

The goal of graph-based SSL is to infer $\boldsymbol{Y}_{miss}$ based on $(\boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})$.

- **Generation process.** The generation process can be illustrated by the following factorization of the joint distribution:

$$p(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{G}) = p(\boldsymbol{G}|\boldsymbol{X}, \boldsymbol{Y})p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X})$$

  where the first and second terms are parameterized by $\boldsymbol{\theta}$: $p_{\boldsymbol{\theta}}(\boldsymbol{G}|\boldsymbol{X}, \boldsymbol{Y})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{Y}|\boldsymbol{X})$.

A flexible generative framework for graph-based SSL (2)

- **Model inference.** To infer the missing outcomes $\boldsymbol{Y}_{miss}$, we would need the posterior distribution $p_{\boldsymbol{\theta}}(\boldsymbol{Y}_{miss}|\boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})$ that is usually intractable.
  $\rightarrow$ Introduce the recognition model $q_{\boldsymbol{\phi}}(\boldsymbol{Y}_{miss}|\boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})$ (variational principles)
- **Model learning.**

$$\log p(\boldsymbol{Y}_{obs}, \boldsymbol{G}|\boldsymbol{X}) \geq -\mathcal{L}_{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})$$

where the lower bound is defined as:

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{Y}_{miss}|\boldsymbol{X},\boldsymbol{Y}_{obs},\boldsymbol{G})}\Big[\log p_{\boldsymbol{\theta}}(\boldsymbol{Y}_{miss}, \boldsymbol{Y}_{obs}, \boldsymbol{G}|\boldsymbol{X}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{Y}_{miss}|\boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})\Big]$$

Optimization:

$$\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}} = \arg\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G})$$

A flexible generative framework for graph-based SSL (3)

- **Latent space models (LSM).** Based on the conditional independence assumption of edges:

$$p_{\theta}(\boldsymbol{G}|\boldsymbol{X}, \boldsymbol{Y}) = \prod_{i,j} p_{\theta}(e_{ij}|\boldsymbol{X}, \boldsymbol{Y})$$

  where:

$$p_{\theta}(e_{ij}|\boldsymbol{X}, \boldsymbol{Y}) = p_{\theta}(e_{ij}|\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{x}_j, \boldsymbol{y}_j)$$

  is modeled by a logistic regression.

- **Supervised loss.** Additional term with weight hyper-parameter η for the approximate posterior model:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathcal{L}_{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{X}, \boldsymbol{Y}_{obs}, \boldsymbol{G}) - \eta \cdot \log q_{\boldsymbol{\phi}}(\boldsymbol{Y}_{obs}|\boldsymbol{X}, \boldsymbol{G})$$

Standard benchmark

Table 1: Summary of benchmark datasets.

Dataset	# Classes	# Nodes	# Edges	Avg. 2-Neighborhood Size
Cora	7	2,708	5,278	35.8
Pubmed	3	19,717	44,324	59.1
Citeseer	6	3,327	4,552	14.1

Table 2: Classification accuracy under the standard benchmark setting. The upper block lists the discriminative baselines. The lower block lists the proposed variants of G^3NN. The **bold** marker denotes the best performance on each dataset. The underline marker denotes that the generative model outperforms its discriminative counterpart, e.g., LSM-GCN outperforms GCN; and the asterisk (*) marker denotes the difference is statistically significant by a t-test at significance level 0.05. The ($\pm$) error bar denotes the standard deviation of the test performance of 10 independent trials.

	Cora	Pubmed	Citeseer
MLP	0.583 ± 0.009	0.734 ± 0.002	0.569 ± 0.008
GCN	0.815 ± 0.002	$\mathbf{0.794} \pm 0.004$	0.718 ± 0.003
GAT	0.825 ± 0.005	0.785 ± 0.004	0.715 ± 0.007
LSM_GCN	0.825 ± 0.002*	0.779 ± 0.004	0.744 ± 0.003*
LSM_GAT	$\mathbf{0.829} \pm 0.003$	0.776 ± 0.007	0.731 ± 0.005*
SBM_GCN	0.822 ± 0.002*	0.784 ± 0.006	$\mathbf{0.745} \pm 0.004$*
SBM_GAT	0.829 ± 0.003	0.774 ± 0.004	0.740 ± 0.003*

Missing-edges setting

Setting: Remove all the edges of the test nodes from the graph.

Table 3: Classification accuracy under the missing-edge setting. The **bold** marker, the underline marker, the asterisk (*) marker, and the (±) error bar share the same definitions in Table 2.

	Cora	Pubmed	Citeseer
MLP	0.583 ± 0.009	0.734 ± 0.002	0.569 ± 0.008
GCN	0.665 ± 0.007	0.746 ± 0.004	0.652 ± 0.005
GAT	0.682 ± 0.004	0.744 ± 0.006	0.642 ± 0.004
LSM_GCN	0.711 ± 0.005*	**0.766** ± 0.006*	0.704 ± 0.002*
LSM_GAT	0.710 ± 0.007*	0.766 ± 0.004*	0.691 ± 0.005*
SBM_GCN	**0.718** ± 0.004*	0.762 ± 0.005*	**0.716** ± 0.004*
SBM_GAT	0.716 ± 0.007*	0.761 ± 0.005*	0.709 ± 0.008*

Reduced-labels setting

Setting: Drop half of the training labels for each class compared to the standard benchmark setting.

Table 4: Classification accuracy under the reduced-label setting. The **bold** marker, the underline marker, the asterisk (*) marker, and the (±) error bar share the same definitions in Table 2.

	Cora	Pubmed	Citeseer
MLP	0.498 ± 0.004	0.674 ± 0.005	0.493 ± 0.010
GCN	0.750 ± 0.003	**0.724** ± 0.005	0.666 ± 0.003
GAT	0.771 ± 0.004	0.711 ± 0.006	0.675 ± 0.005
LSM_GCN	0.777 ± 0.002*	0.709 ± 0.003	0.691 ± 0.005*
LSM_GAT	0.792 ± 0.004*	0.699 ± 0.003	0.691 ± 0.004*
SBM_GCN	0.780 ± 0.002*	0.710 ± 0.004	**0.703** ± 0.006*
SBM_GAT	**0.796** ± 0.008*	0.699 ± 0.003	0.698 ± 0.003*