Evaluating Autoencoder Methods for Building a Molecule Graph Autoencoder

Amelia Woodward {ameliawd}

Mentors: Keiran Thompson [1,2] and Todd Martinez [1,2]

[1] Stanford PULSE Institute, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

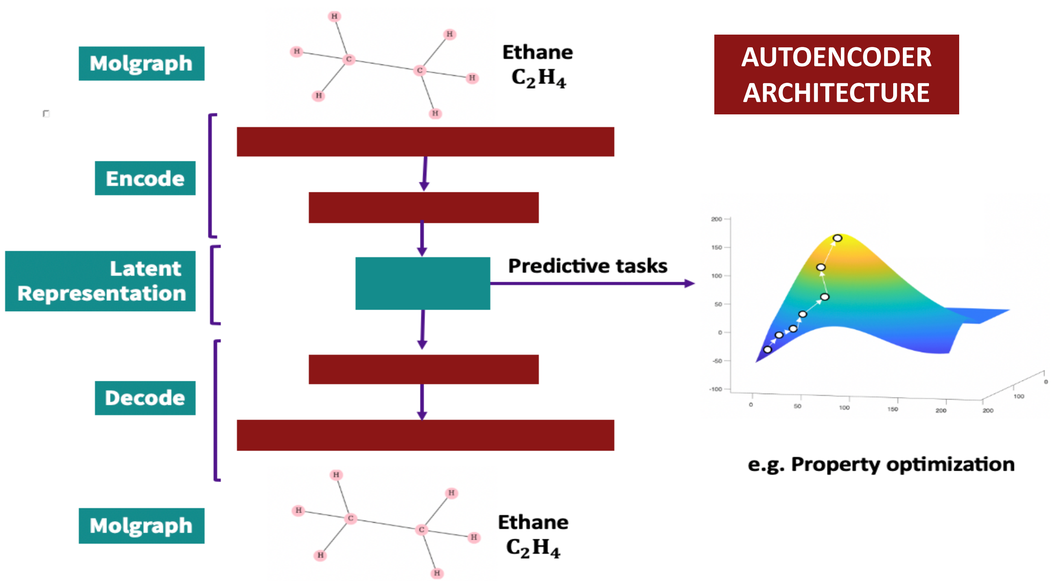
[2] Department of Chemistry and The PULSE Institute, Stanford University, Stanford, California 94305, USA

MOTIVATION

Molecule autoencoders can be used in diverse machine learning tasks in chemistry from drug design (e.g. property optimization) to synthesis planning (e.g. link prediction).

Molecule graphs **(Molgraphs)** hold rich spatial and contextual information about a molecule. It is therefore desirable to build an autoencoder from Molgraphs, yet this remains an open area of research in chemoinformatics.

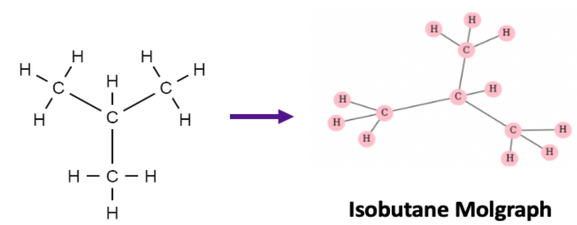
Project Goal: assess existing graph autoencoding methods for use in building a molgraph autoencoder.



DATASET AND GRAPH AUTOENCODING METHODS

Data **GDB13 Database:**

- Contains feasible molecules containing only C,H,N,O,S
- All molecules have ≤ 13 Carbons.
- Data Pre-Processing:



Isobutane Text Representation
'CC(C)C'

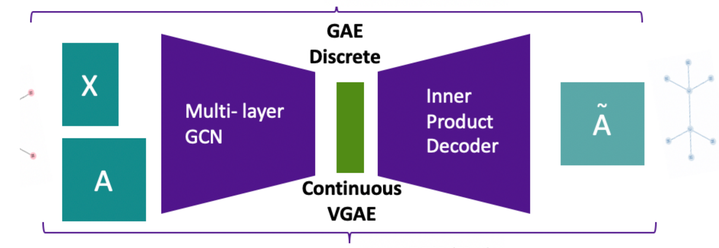
Isobutane Molgraph
G = (V, E)
V = one-hot encoded atoms
E = bond adjacency by covalent distance (Networkx, RDKIT)

Graph Autoencoder: a feedforward neural network. Given some graph *G* we wish to *encode* G to some latent space X, then *decode* to recover *G.*

Molgraph Input: Let *G* be a Molgraph:
G = (X, A) or (F, A)
V = atom feature matrix,
E = bond adjacency matrix

GAE/VGAE [4]

GAE Recon Loss $E_{q(z|X,A)}[\log(p(A|Z)]$

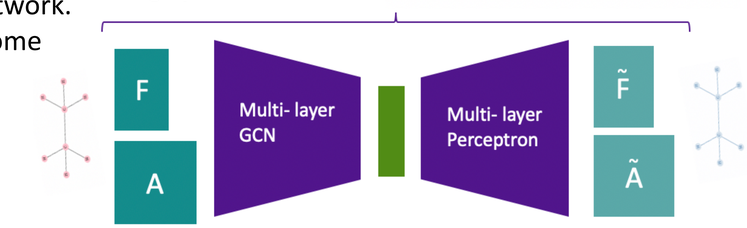


VGAE $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, \quad \mu = GCN_\mu(X,A), \quad \log\sigma = GCN_\sigma(X,A)$

Recon Loss $E_{q(z|X,A)}[\log(p(A|Z)] - KL(q(Z|X,A)||p(Z)).$

Adapted graphVAE [5]

graphVAE Recon Loss : $-logp(G|z) = -\lambda_A \log p(A'|z) - \lambda_F \log p(F'|z)$



EXPERIMENTS AND RESULTS

Experiments — Using processed Molgraphs from GDB13

Experiment 1:
- Train: 10K, Val: 1K, Test: 1K
- 400 Epochs
- ≤ 4 Carbons
- ≤ 6 Carbons
- ≤ 8 Carbons

Model	Metric	≤ 4Cs	≤ 6Cs	≤ 8Cs
GAE	RL Test	1.20	1.13	1.13
	L2	6.4	8.5	8.5
	GED	154	228	227
VGAE	RL Test	2.00	1.98	1.98
	L2	6.7	8.5	8.6
	GED	154	228	227
graphVAE (adapted)	RL Test	-2.75	-2.95	-2.94
	L2	14.2	17.2	17.1
	GED	155	228	227

Experiment 2:
- Train: 100K, Val: 10K, Test: 10K
- 15 - 30 Epochs
- ≤ 8 Carbons
- 16 channels
- 32 channels
- 2 Conv Layers
- 3 Conv Layers

Model	Metric	16 Channel 2 Conv	32 Channel 2 Conv	16 Channel 3 Conv	32 Channel 3 Conv
GAE	RL Test	1.04	1.05	1.09	1.12
	L2	11.8	12.0	12.3	12.3
	GED	228	237	235	234
VGAE	RL Test	1.89	2.46	1.95	2.55
	L2	11.9	11.9	12.2	12.4
	GED	239	234	233	238
graphVAE (adapted)	RL Test	-2.95	-2.9	-2.95	-2.93
	L2	17.2	16.5	17.2	16.6
	GED	233	233	235	235

Metrics

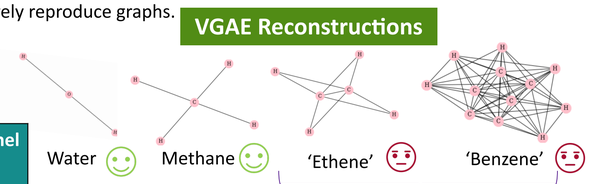
Reconstruction Loss
$$E_{q(z|V,E)}[\log(p(E|Z)]$$
$$E_{q(z|V,E)}[\log(p(E|Z)] - KL(q(Z|V,E)||p(Z))$$
$$-logp(G|z) = -\lambda_E \log p(E'|z) - \lambda_V \log p(V'|z)$$

Average L2 - loss
$$loss_{L2} = \frac{1}{n}\sum_{i=1}^{n} ||A_n - \bar{A}_{thresh_n}||_2.$$

Average Graph Edit Distance
$$GED(g_1, g_2) = \min_{e_1,e_2,\ldots,e_k \in \mathcal{P}(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$
$$GED_{average} = \frac{1}{n} GED_n$$

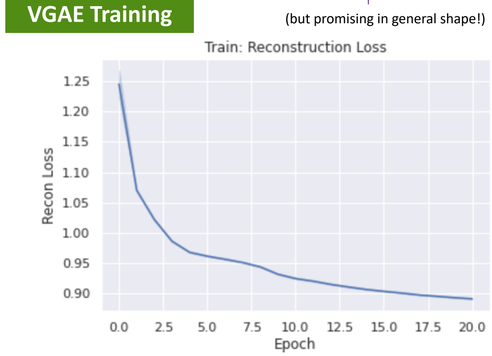
VGAE: Reconstructs methane and very simple organic molecules, almost always overdraws edges for larger molecules. GAE also has similar results, but we want a continuous representation, so favor VGAE. **graphVAE (adapted)** currently does not effectively reproduce graphs.

VGAE Reconstructions



(but promising in general shape!)

VGAE Training



OUTCOMES AND NEXT STEPS

1. **VGAE appears to be more flexible and have significantly better Molgraph reconstruction** outcomes (even when not fully converged) than adapted graphVAE. Consider adding node and edge attribute features to this model to explore further development.

2. Positively, all models are learning during training in that their reconstruction loss is decreasing. To optimize, **conduct comprehensive hyperparameter search** to determine optimal learning rates, explore adding many more convolutional layers. Especially, uncover why graphVAE saturates after 1-2 epochs (may lead to significantly improved results).

   All models tend to over-draw edges and require high thresholds in output adjacency matrix to recover 'molecule-like' outputs. We should explore why this occurs/what thresholding techniques make the most sense with these models.

4. We hypothesize that **adding edge weights containing bond order information may drastically help the model learn**. For instance bond order information contains tells the model whether a carbon should be connecting to 2,3 or 4 other atoms. We will try training with bond orders.

REFERENCES

[1] Wang, L., Titov, A., McGibbon, R., Liu, F., Pande, V., & Martínez, T. (2014). Discovering chemistry with an ab initio nanoreactor. Nature Chemistry, 6(12), 1044-1048. doi: 10.1038/nchem.2099.

[2] Gómez-Bombarelli, R. et al. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. ACS Central Science, 4(2), pp.268-276.

[3]Woodward, A. (2019) Machine Learning on Chemical Reaction Networks: Summer Research Update. (My summer research last year)

[4] Kipf and Welling. (2016). Variational Graph Autoencoders. arXiv:1611.07308v1

[5] Simonovsky and Komadakis. (2018) graphVAE: Towards the Generation of Small Graphs Using Variational Autoencoders.arXiv:1802.03480v1

[6] Reymond Research Group. (2007) GDB Database. University of Bern http://gdb.unibe.ch/