Molecular Design Model Comparison in Deep Evolutionary Learning

Liam McDevitt
Department of Computer Science
Brock University
St. Catharines, ON
lm15ue@brocku.ca

Abstract

In this paper, we compare two recently successful molecular design model methods, Fragment Variational Autoencoder (FragVAE) and Junction Tree Variational Autoencoder (JT-VAE), in a newly proposed deep evolutionary learning (DEL) method for molecular drug design. DEL does this by combining two well-researched topics, multi-objective evolutionary computation and deep generative models. Computational molecular drug design methods would be incomplete without a way to model molecules. The molecules in DEL are models by a Variational Autoencoder (VAE), and it was hypothesized that a change in the type of VAE would affect DEL's overall performance. The original DEL implementation used a FragVAE, and to compare against a JT-VAE, a separate JT-VAE-based DEL had to be created. Experiments on the public MOSES dataset indicate further research and experiments are required to come up with a definite answer on whether a FragVAE-based DEL outperforms a JT-VAE-based DEL or vice versa.

1 Introduction

Combating against rare and complicated diseases and deadly cancers is a tremendous feat for researchers worldwide. Designing and discovering drugs to aid in this battle is a timely and expensive task. Help can be found in developing and designing valuable new medicines through the field of computational intelligence. Although computational intelligence has provided vast achievements in this field, there is still a need for powerful search algorithms and impartial feature theories for accurately representing molecules with their corresponding receptors. In recent years there has been a boom in AI and data scientists tackling this problem due to the exponential advancements in technology allowing them to develop efficient, well-informed design and search strategies (Chen et al., 2018; Gromski et al., 2019).

In recent years within the realm of molecular generation, state-of-the-art architectures and representation theories have been proposed. In machine learning, there are two standard methods for representing a molecule. The first and most primitive method is to describe the molecule as a string of ASCII characters known as a simplified molecular-input line-entry system (SMILES) (Weininger, 1988). Natural language processing (NLP) is used for either supervised or unsupervised learning in conjunction with SMILES. The second method represents its molecule as an undirected graph which is applied to graph (convolutional) neural networks (Duvenaud et al., 2015).

Molecular generation is possible by adopting generative autoencoders (e.g. variational autoencoder (VAE) (Kingma & Welling, 2014)) to learn on either of the presented molecular representation methods: SMILES strings (Romez-Bombarelli et al., 2018) or molecular graphs (Simonovsky & Komodakis, 2018; Jin et al., 2019; 2020). Generative autoencoders are a great asset in generating new discrete molecular objects with favoured properties since they can map molecules to a continuous

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

latent space where the landscape allows us to see and organize based on their individual properties more efficiently. Nonetheless, these molecule representation methods are far from perfect and often lead to severe complications. Suppose SMILES strings are used in VAE as a molecular representation. In that case, the model can often generate invalid structures leading to false molecules. There is inequality among the tokens within the embedding, and almost indistinguishable molecules have remarkably contrasting SMILES strings. If a graph representation for a molecule is used in VAE, a complicated obstacle arises in constructing a robust graph decoder.

Two new promising methods were introduced as an upgrade compared to previous molecular representation strategies within the past two years, FragVAE (Podda et al., 2020) and JT-VAE (Jin et al., 2019). FragVAE is based on a Fragment-Based Drug Design paradigm where molecules are not generated atom by atom but instead generated fragment by fragment. JT-VAE converts initial SMILES strings into a graph representing a molecule where each graph comprises smaller prevalidated subgraphs. A significant benefit both techniques provide in comparison to previous molecular representation strategies is molecule validity. Since molecules are generated by fragments (FragVAE) or by combining subgraphs (JT-VAE), each molecule comprises only valid sub-components, overall increasing generation validity. The main difference between these two methods is that JT-VAE is a graph-based modelling technique and FragVAE works by decomposing SMILES string encodings directly.

A newly proposed method combining evolutionary computation and deep generative models known as deep evolutionary learning (DEL) has shown great promise in molecular design (Li et al., 2021). For DEL, it is possible to use different molecule representations, and it is hypothesized that other representations will affect the overall performance of DEL. DEL incorporated a FragVAE model; although this showed commendable results, this is the only molecule representation technique tested. Hence, trying a JT-VAE-based DEL in comparison to a FragVAE-based DEL may present some interesting findings.

In this paper, a FragVAE-based DEL will be compared to a JT-VAE-based DEL on the ZINCMOSES dataset. The method section will cover FragVAE, JT-VAE, and DEL. The used data source will be given and explained in the experiment section, and the experimental procedures, processing steps, and model hyperparameters. The experiment section will also report upon and discuss the results. The conclusion section will provide a short conclusion with future work and insights.

2 Method

This paper does not attempt to produce a new method but instead compares a change to a component in an already existing method. Since VAE in DEL is said to be abstractable, different VAE's may affect the performance. In the original DEL, a FragVAE is used. This paper attempts to replace FragVAE in the DEL model with JT-VAE and perform experiments on both a FragVAE-based DEL and a JT-VAE-based DEL to observe DEL's performance concerning an altercation in the VAE used. In essence, we're testing DEL based on two different methods for representing molecules. The following subsections in this section will cover JT-VAE, FragVAE, and DEL in more detail.

2.1 Junction Tree Variational Autoencoder

A Junction Tree Variational Autoencoder was introduced in 2019 by Jin et al. for Molecular Graph Generation. The writers introduced a way of generating molecular graphs to ensure chemical validity at each incremental expansion of a molecule's generation. This approach is an improvement on SMILES representations and an improvement on standard generating graphs node by node. SMILES cannot capture molecule similarity, and the validity of the models is much more apparent on a graph than on a linear representation. Learning smooth embedding of the molecules is almost impossible for SMILES since strings are maybe very different for very similar molecules. The node by node approach of other graph-based models is not feasible from a molecular generation standpoint because generating atom by atom is a recipe for invalid molecules, especially at intermediary steps. We cannot be sure if the molecule is valid until the end of its generation.

The JT-VAE approach first starts by generating junction trees which each represent a valid molecular subgraph. These junction trees are extracted using tree decomposition from the training set. Once all of the valid subgraphs are generated, the next step is to put them together to form a molecular graph.

When sampled from a prior distribution, Jin et al. demonstrated that their approach produced 100% valid molecules, in turn vastly outperforming their competition at the time.

2.2 Fragment-based Molecular Modelling by FragVAE

A deep generative model for fragment-based molecule generation (FragVAE) was introduced in 2020 by Podda et al., inspired by the well-known fragment-based drug design (FBDD) paradigm. The idea behind this paradigm is to combine fragments, smaller well put together molecules that are easily bindable, to form more sophisticated compounds. Hence, the model proposed models molecules with the same basic approach of generating molecules fragment by fragment instead of atom by atom to create preferable molecules.

The FragVAE approach starts by creating an ordered sequence of fragments by breaking down molecules from a given dataset. The breakdown of molecules into fragments is done by the Breaking of Retrosynthetically Interesting Chemical Substructures (BRICS) algorithm by applying breakage rules to keep sound molecules. The fragmentation algorithm goes through a SMILES encoding, and when BRICS finds a breakable bond, it is broken in two by applying a matching chemical reaction. This deconstruction of the molecule is done left to right in a recursive fashion allowing this to be done in reverse. Therefore, this can match closely to the encoder-decoder procedure of a VAE.

2.3 Deep Evolutionary Learning (DEL)

A deep evolutionary learning (DEL) strategy was proposed by Li et al. in 2021 for molecular drug design by bringing together deep generative models and multi-objective evolutionary computation (EC) in the latent representation space of molecules. DEL is built upon their theoretical innovation that EC methods can be used in relating deep versions. The DEL strategy is comprised of the following steps: (1) The original training data is used to pre-train a VAE and a multilayer perceptron neural network (MLP) for the first evolutionary generation. If it's not the first generation, then they are pre-trained from the samples in the previous population; (2) The VAE encoder is used to project the new samples into the latent space; (3) The individual latent representations of the samples within the latent space undergo evolutionary operations (i.e., selection, crossover, and mutation) based on crowding distances and non-dominated ranking contributing to the multi-objectivity; (4) New samples are generated from the VAE decoder based on the newly latent representation individuals after the evolutionary operations; (5) RDKit (Landrum, 2006) is used to obtain the chemical properties of the newly formed individuals; (6) Favourable generated samples makeup the new population; (7) Steps (2-6) are repeated for a set number of generations; (8) The final population contains our best-found individuals and is returned. Most of DEL's advantages in comparison to other state-of-the-art methods is the evolutionary aspect. Using EC on the continuous latent representation allows for smooth and efficient exploration of the encoded search space. EC methods are great at optimization, and the multi-objective methods DEL uses helps keep selection diverse and competitive.

3 Experiment

3.1 Code Sources

Both the code source for the FragVAE-based DEL and the JT-VAE-based DEL can be found at the provided link: FragVAE-DEL-CODE & JT-VAE-DEL-CODE. The original DEL implementation can be found at this provided link: DEL. The original JT-VAE implementation can be found at this provided link: JT-VAE.

3.2 Data Source

The processed data from MOSES (Polytkovskiy et al., 2020) was used to compare a FragVAE-based DEL and a JT-VAE-based DEL. The MOSES dataset was created to help compare generative models in the molecular space. The models may learn on this dataset and generate new molecules with similar characteristics.

3.3 Processing Steps

For both the FragVAE-based DEL and the JT-VAE-based DEL, the MOSES data is already processed within the DATA/ZINCMOSES/PROCESSED directory location for both respective implementations. The original full training and testing data is called trainFull.smi and testFull.smi in the same location. The ones used for this paper's experiments were smaller with 250 samples called train.smi and test.smi due to system limitations. There are the following README files in most implementation directories for any new datasets, statistics, graphing, etc.

It should be noted that the datasets for FragVAE and JT-VAE had different formats. Hence, processing the data for both to work for DEL was relatively complex. The dataset for FragVAE had multiple columns while JT-VAE just had one SMILES string column. Since the dataset for FragVAE is tightly woven into DEL, the JT-VAE dataset needed to be converted to a FragVAE so DEL can do its scoring and operations. Then it needed to be converted back to JT-VAE could train properly.

Experiments were conducted on a Virtual Machine, thus no GPU, running 6 GB of RAM and four processors. Using this hardware set-up for the experiments was very limiting on the number of tests and the size of the tests that could be conducted. The original DEL parameters have been drastically reduced to finish running within the allotted experimental time frame.

3.4 Model Hyperparameters

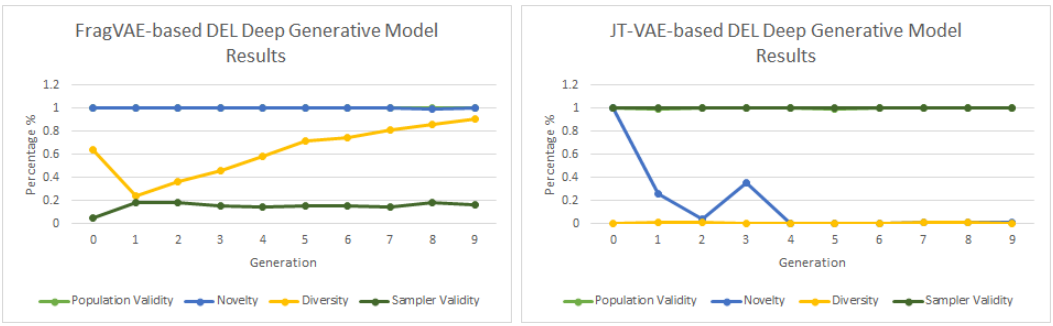
DEL's most important evolutionary parameters were: population size = 250, number of generations = 10, mutation rate = 1% and crossover type = linear. Other important parameters include: learning rate = 0.0001, step size = 4, γ = 0.8, batch size = 128, hidden size = 128, hidden layer = 2, and latent size = 32. Many of FragVAE and JT-VAE parameters go hand in hand. Both implementations were run once due to computing and time limitations. For a full rundown of all used configuration parameters used please see each implementations completed RUNS folder with specific runs config/params.json file. Both train and test .smi files for FragVAE and JT-VAE testing are the same with 250 samples each.

3.5 Experimental Procedures

For both FragVAE-based DEL and JT-VAE-based DEL, the runscript5_1.sh file was run from the root directory containing the same run command for both versions. However, the configuration files differ slightly, but that is only due to different parameters being required for JT-VAE compared to FragVAE. One run was conducted for both, and the data acquired from the RUNS folder for each particular run will be used to draw our conclusions.

3.6 Results & Discussions

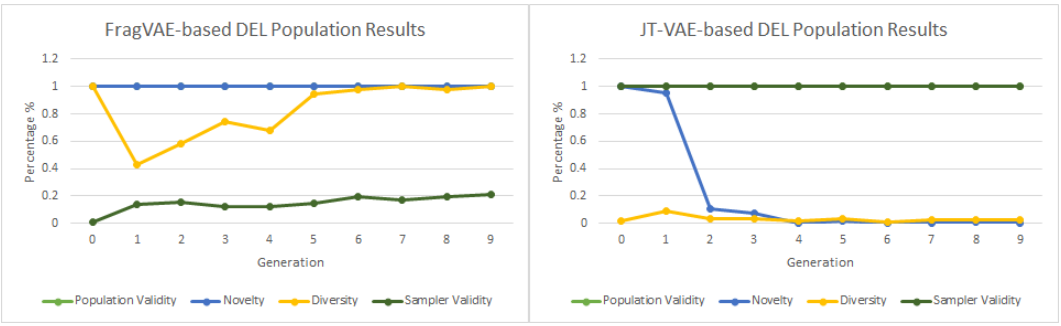
It should be noted that it is highly probable that the results obtained are faulty due to implementation and system integration challenges. In addition, due to system limitations, there is no ground to make any significant comparisons on the results since only one run was conducted and with the smallest parameter choices compared to the original implementation of DEL. The subsequent plots were made from the resulting output the initial DEL implementation generates after program execution for both the FragVAE-based DEL and the JT-VAE-based DEL.



(a) FragVAE-based DEL (b) JT-VAE-based DEL

Figure 1: The deep generative model's population validity, novelty, diversity, and sampler validity results.

Fig. 1a and Fig. 1b show the deep generative model's (DGM) population validity, novelty, diversity, and sampler validity results over 10 runs. It can be seen from the plots that the novelty for the DGM remains consistent in the Frag-VAE-based DEL at around 100%, but the novelty for the JT-VAE-based DEL drops off quickly, eventually converging at 0%. The diversity for the Frag-VAE-based DEL improves over the generations but remains almost non-existent for the JT-VAE-based DEL. The population validity for the FragVAE-based DEL is covered by the novelty and also remains around 100%. The JT-VAE-based DEL has the same case for its population validity. For the FragVAE-based DEL, the sampler validity steadily climbs to 20%, but the JT-VAE-based DEL's sampler hovers around 100 %.



(a) FragVAE-based DEL (b) JT-VAE-based DEL

Figure 2: The population's population validity, novelty, diversity, and sampler validity results.

Fig. 2a and Fig. 2b show the populations population validity, novelty, diversity, and sampler validity results over 10 runs. It can be seen from the plots that the novelty for the DGM remains consistent in the Frag-VAE-based DEL at around 100%. Still, the novelty for the JT-VAE-based DEL declines slightly for the first generation but then drops off quickly, eventually converging to 0% near the 6th generation. The diversity for the Frag-VAE-based DEL improves overall over the generations converging at 100% but has an initial drop off at the first generation and remains almost non-existent JT-VAE-based DEL. The population validity for the FragVAE-based DEL is covered by the novelty and also remains around 100%. The JT-VAE-based DEL has the same case for its population validity. For the FragVAE-based DEL, the sampler validity steadily climbs to 20%, but the JT-VAE-based DEL's sampler hovers around 100 %.

As discussed earlier, these results are not a good indication of DEL's performance concerning the two different molecular modelling methods since the integration of JT-VAE into DEL may not have been successful, thus rendering any results mute. Further time should be spent on a robust integration scheme of different VAEs into DEL's framework.

4 Conclusions & Future Work

This paper attempted to integrate a JT-VAE into DEL to replace FragVAE to compare the differences in DEL's performance based on different state-of-the-art molecular representation methods. The integration process of JT-VAE into DEL was cumbersome and involved complicated rewiring due to many DEL-specific dependencies concerning FragVAE. The results found from the experiments are not enough to scientifically justify using one method over the other besides the fact that the JT-VAE-based DEL implementation we provided may not be correct. Therefore, it would be beneficial to use a FragVAE-based DEL situated solely on the implementation already being done as a general recommendation. In conclusion, further work should be conducted to recreate DEL from the ground up where the used VAE implementation is completely abstractable to test a FragVAE-based DEL, a JT-VAE-based DEL, including other possible VAE methods in more detail to draw significant comparable results.

References

[1] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," J. Chem. Inf. Model., vol. 28, no. 1, pp. 31–36, Feb. 1988, doi: 10.1021/ci00057a005.

[2] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv:1312.6114 [cs, stat], May 2014, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1312.6114.

[3] D. Duvenaud et al., "Convolutional Networks on Graphs for Learning Molecular Fingerprints," arXiv:1509.09292 [cs, stat], Nov. 2015, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1509.09292.

[4] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," Drug Discovery Today, vol. 23, no. 6, pp. 1241–1250, Jun. 2018, doi: 10.1016/j.drudis.2018.01.039.

[5] R. Gómez-Bombarelli et al., "Automatic chemical design using a data-driven continuous representation of molecules," ACS Cent. Sci., vol. 4, no. 2, pp. 268–276, Feb. 2018, doi: 10.1021/acscentsci.7b00572.

[6] M. Simonovsky and N. Komodakis, "GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders," arXiv:1802.03480 [cs], Feb. 2018, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1802.03480.

[7] P. S. Gromski, A. B. Henson, J. M. Granda, and L. Cronin, "How to explore chemical space using algorithms and automation," Nat Rev Chem, vol. 3, no. 2, pp. 119–128, Feb. 2019, doi: 10.1038/s41570-018-0066-y.

[8] W. Jin, R. Barzilay, and T. Jaakkola, "Junction Tree Variational Autoencoder for Molecular Graph Generation," arXiv:1802.04364 [cs, stat], Mar. 2019, Accessed: Apr. 26, 2021. [Online]. Available: http://arxiv.org/abs/1802.04364.

[9] W. Jin, K. Yang, R. Barzilay, and T. Jaakkola, "Learning Multimodal Graph-to-Graph Translation for Molecular Optimization," arXiv:1812.01070 [cs, stat], Jan. 2019, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1812.01070.

[10] W. Jin, R. Barzilay, and T. Jaakkola, "Multi-Objective Molecule Generation using Interpretable Substructures," arXiv:2002.03244 [cs, stat], Jul. 2020, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/2002.03244.

[11] M. Podda, D. Bacciu, and A. Micheli, "A Deep Generative Model for Fragment-Based Molecule Generation," arXiv:2002.12826 [cs, stat], Feb. 2020, Accessed: Apr. 26, 2021. [Online]. Available: http://arxiv.org/abs/2002.12826.

[12] D. Polykovskiy et al., "Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models," arXiv:1811.12823 [cs, stat], Oct. 2020, Accessed: Apr. 28, 2021. [Online]. Available: http://arxiv.org/abs/1811.12823.

[13] Y. Li, H. K. Ooi, and A. Tchagang, "DEEP EVOLUTIONARY LEARNING FOR MOLECULAR DESIGN," p. 33, 2021.