NOVEL JUNCTION TREE NEURAL NETWORK BASED SIMILARITY MEASURE

Chirstopher Falholt Elvebakken, Tobias Wiik Thalbitzer, Mikkel Staby
{s165790, s153900,s134388}@student.dtu.dk

Technical University of Denmark - DTU

ABSTRACT

Deep generative models are innovating the world of drug design, as they can assist in the search of new molecules by generating new molecules with similar chemical features. In this project, the state-of-the-art model for molecular graph generation (a junction tree variational autoencoder) is trained and employed on the ZINC dataset. The model works by decomposing molecules to a junction tree and then encodes and decodes both the molecular graph and the junction tree.
To assess similarity of molecules, previous measures such as the Tanimoto/Jaccard similarity are used. A recent paper suggest using the euclidean distance in latent space as a similarity measure. This project proposes a new revised similarity measure based on the euclidean distance. This similarity measure consists of using the euclidean distance in latent space conditioned that the molecule are within the distribution generated by the variational autoencoder.
Finally, we proceed to show that euclidean based distances in latent space are more representative when assessing similarity of chemical features rather than atomwise similarity.

***Index Terms*—** Generative deep learning, molecular graph generation, variational autoencoder, junction tree variational autoencoder, junction tree neural network, molecular similarity

1. INTRODUCTION

The world of *de novo* molecular design is complex and challenging for drug designers and biochemists, as molecular combinations can reach 10^{60} possibilities, which results in an infeasibility of creating all molecules [1].
Due to the vast amount of possibilities, finding target molecules with desired chemical properties are difficult.
The introduction of generative modelling, more specifically, variational autoencoders (VAE's) into this field of study has innovated the design process, as it allows for an *inverse design* structure, where chemical properties are chosen from a functional space (called the latent space) and a molecule with these properties are then generated by the VAE-decoder [2].
Using this method, drug designers can easily construct new molecules with desired features as shown in Fig. 1.

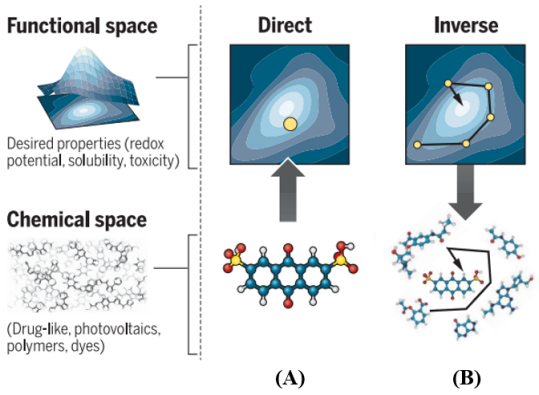


Fig. 1. Graphical illustration of the *direct design* structure, where desired properties are found through simulations or experiments **(A)** compared to how VAE's use *inverse design* structure **(B)** to discover chemical structures with the wanted properties. Partially reprinted from [2].

When working with chemical molecules and machine learning, numerous representations of the molecules are possible which allows for neural networks and other models to interpret the data. Of these representations the most commonly used are: fingerprint vectors, SMILES strings (simple text encodings of the chemical structure), potential energy functions, graphs with atom and bond weights, the coulomb matrix, bag of bonds and fragments, 3D geometry with associated atomic charges and electronic densities [2]. Due to its simplicity, the SMILES representation is a popular representation [3]. Although the SMILES data format is simple, it has many issues as fairly similar molecules can have completely different SMILES representations, which will result in large issues when modelling the data. As chemical structures consist of atoms interconnected by molecular bonds, using molecular graphs with atom and bond weights is used in this project. This increases the complexity
This report examines a paper published by Jin et al. [4] on the topic of directly encoding molecular graphs instead of the popular choice, SMILES, using a junction tree variational

auto enconder (JTVAE). This direct usage proves advantageous in several ways and outperforms previous state-of-the-art models. We seek to explore the testing of similarity between the molecules by checking if their latent representation is out of distribution, followed by a measurement of its euclidean distance to check for similarity. It will first be done on ZINC data set, then followed by QM9.

2. METHODS

2.1. Dataset and Data Preprocessing

2.1.1. ZINC Dataset

The datasets used for this project is the ZINC dataset from Kusner et al. [5]. The dataset concists of 250,000 different molecules in SMILES format, however, only 10,000 datapoints will be used to reduce computation times.
The dataset is converted into molecular graphs through deterministic mappings using the RDKit package [6].
Furthermore, a few molecules from the ZINC dataset has been displayed in Fig. 2.

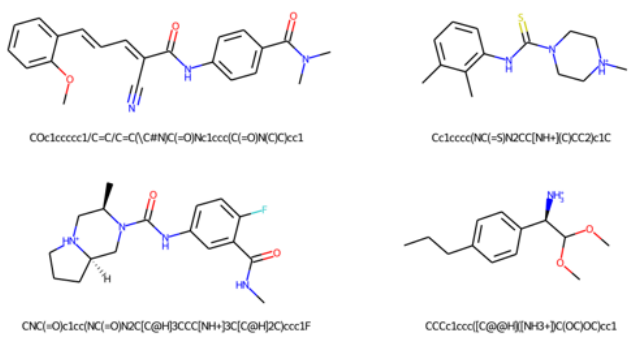


Fig. 2. 4 randomly selected molecules from the ZINC dataset and their corresponding SMILES representations.

2.1.2. QM9 Dataset

The QM9 dataset consists of geometric, energetic, electronic and thermodynamic properties for 133,885 stable small organic molecules. The JTVAE model was also trained on the QM9 dataset. After 7 days of computation, the training of the model was abandoned due to unavailability to meet deadlines. It was then chosen to further work only with the ZINC dataset due to the computational limitations.

2.2. Input Representation

The VAE encodes two different sets of latent space by encoding two different representations of the molecule. The first one is a molecular graph, and the second is a junction tree.

2.2.1. Vocabulary Generation

Through the RDkit package [6], a vocabulary of sub-graphs are generated from the dataset. It searches through our dataset and computes every ring structure, individual atoms and bonds included in the molecules of the ZINC data set.

2.3. Junction Tree Variational Autoencoder

The JTVAE is an extension to the variational autoencoder and consists of 5 major components. These major components are the tree decomposition (briefly summarized in section 2.3.1), a graph encoder and decoder and a tree encoder and decoder. Furthermore, Fig. 3 shows in brief how the JTVAE works by decomposing the molecule into a junction tree, followed by a encoding of both the molecular graph and the junction tree to a latent space, $\mathbf{z}_G$ and $\mathbf{z}_T$.
The molecule can then be reconstructed by using the tree decoder to decode the latent representation, $\mathbf{z}_T$, into a junction tree, which is then transformed back to the original molecule using the graph decoder and the latent representation $\mathbf{z}_G$.

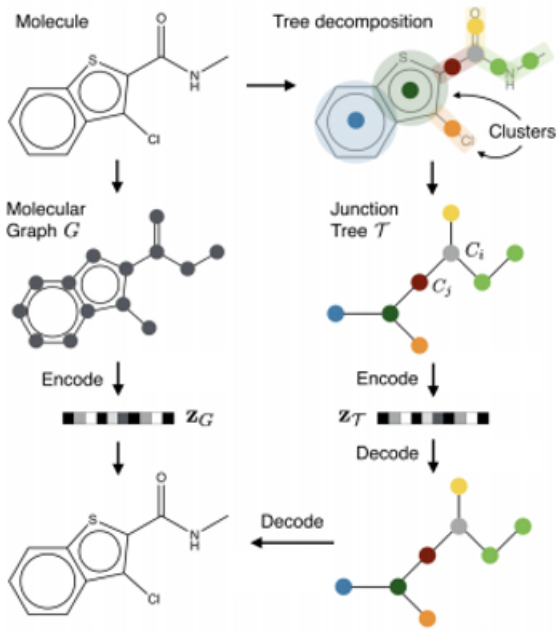


Fig. 3. A brief overview of how the JTVAE works. The molecular graph is decomposed into to its junction tree. The molecular graph and the junction tree are then decoded to respective latent embeddings, $\mathbf{z}_G$ and $\mathbf{z}_T$. The latent representation can then be decoded back to its junction tree from $\mathbf{z_T}$. Finally, the molecules in the respective clusters are then found from the graph decoder $p(G|T, \mathbf{z}_G)$. Partially reprinted from [4].

The encoders and decoders contained in the JTVAE will be briefly described in the subsections below.

2.3.1. JTVAE Tree Decomposition

First the molecules will be "tree decomposed", which maps the molecule into a graph, G, consisting of a node set, $\mathcal{V}$ and an edge set $\mathcal{E}$. Here the node set consists of a substructure of the molecule which is contained in the vocabulary, $\mathcal{X}$. The tree decomposition algorithm employs the cluster vocabulary, $\mathcal{X}$, to first find all simple cycles in a given molecule. The simple cycles are then merged together if they have more than 2 overlapping atoms. The cluster graph is then generated by adding "bonds" between neighboring clusters.
This tree decomposition provides chemical feasibility throughout the generation, as it ensures that only valid clusters are added to the junction tree. Furthermore, as a result of 2 clusters having at most 2 atoms in common, it is allowing for efficient inference in the graph decoding phase.

2.3.2. Graph Encoder

The graph encoder works by employing a graph message passing network. Here, each atom of the graph has a feature vector, $\mathbf{x}_a$, which indicates chemical properties, such as atom type and valence. Furthermore, each interconnection of the atoms are described by a feature vector (noted as $\mathbf{x}_{ab}$) which indicates the bond type and 2 hidden vectors v_{ab} and v_{ba} which describes the messages from atoms a and b. The hidden vectors are given as shown in eq. (1).

$$v_{ab} = \text{ReLU}\left(\mathbf{W}_1^g \mathbf{x}_a + \mathbf{W}_2^g \mathbf{x}_{ab} + \mathbf{W}_3^g \sum_{c \in N(a)\setminus b} v_{ca}^{(t-1)}\right) \tag{1}$$

Here v_{ab} is the message in the t-th iteration, which is initially set to 0 and $N(x)$ is the neighbors of x.
After T iterations, all messages are aggregated to the latent vector shown in eq. (2).

$$\mathbf{h}_a = \text{ReLU}\left(\mathbf{U}_1^g \mathbf{x}_a + \sum_{b \in N(a)} \mathbf{U}_2^g v_{ba}^{(T)}\right) \tag{2}$$

The final graph representation is then found as shown in eq. (3).

$$\mathbf{h}_G = \sum_i \mathbf{h}_i / |A| \tag{3}$$

Where $|A|$ is the set of atoms in the molecule. From $\mathbf{h}_G$ the mean, μ_G and variance, σ_G, of the variational posterior distribution can be computed. Finally, $\mathbf{z}_G$ is sampled from a the Gaussian, $\text{N}(\mu_G, \sigma_G)$.

2.3.3. Tree Encoder

The tree encoder employs a message passing network. Each cluster, C_i, achieved from the tree decomposition is transformed to a vector, $\mathbf{x}_i$, using one-hot encoding. Furthermore, interconnections between clusters are associated with two message vectors, $\mathbf{m}_{ij}$ and $\mathbf{m}_{ji}$. A random initialization point is chosen for the decomposition of each molecule. The messages are then propagated in two phases. First a bottom-up phase, where messages are initiated from the leaf nodes and propagated iteratively towards the root of the molecule. This phase is then followed by a top-down phase, where messages are propagated from the root to all the lead nodes.
The messages are updated as shown in eqs. (4)-(8). The messages are calculated by a Gated Recurrent Unit (GRU) [7] adapted to pass messages from the trees.

$$\mathbf{m}_{ij} = (1 - \mathbf{z}_{ij}) \odot \mathbf{s}_{ij} + \mathbf{z}_{ij} \odot \tilde{\mathbf{m}}_{ij} \tag{4}$$

$$\tilde{\mathbf{m}}_{ij} = \tanh\left(\mathbf{W}\mathbf{x}_i + \mathbf{U} \sum_{k \in N(i)\setminus j} \mathbf{r}_{ki} \odot \mathbf{m}_{ki}\right) \tag{5}$$

$$\mathbf{r}_{ki} = \text{Sigmoid}\left(\mathbf{W}^r \mathbf{x}_i + \mathbf{U}^r \mathbf{m}_{ki} + \mathbf{b}^r\right) \tag{6}$$

$$\mathbf{z}_{ij} = \text{Sigmoid}\left(\mathbf{W}^z \mathbf{x}_i + \mathbf{U}^z \mathbf{s}_{ij} + \mathbf{b}^z\right) \tag{7}$$

$$\mathbf{s}_{ij} = \sum_{k \in N(i)\setminus j} \mathbf{m}_{ki} \tag{8}$$

Once the message passing is calculated, the latent representation of the molecules are calculated by aggregating from leaf to root for the messages, in accordance with eq. (9).

$$\mathbf{h}_i = \text{ReLU}\left(\mathbf{W}^o \mathbf{x}_i + \sum_{k \in N(i)} \mathbf{U}^o \mathbf{m}_{ki}\right) \tag{9}$$

The latent representation of the junction tree is then given as the root of the molecule $\mathbf{h}_G = \mathbf{h}_{\text{root}}$.

2.3.4. Tree Decoder

The tree decoder uses a tree structured decoder on a junction tree T from its encoding, $\mathbf{z}_T$. It generates one node at a time by beginning at the root and generates all visited nodes in their depth-first order and backtracks when the node does not have any more children to generate. It uses information from message vectors $\mathbf{h}_{ij}$ from the already visited notes to make new predictions when constructing the tree. The hidden messages are found as shown in eq. (10).

$$\mathbf{h}_{i_t, j_t} = \text{GRU}\left(\mathbf{x}_{i_t}, \{\mathbf{h}_{k,i_t}\}_{(k,i_t) \in \tilde{\mathcal{E}}_t, k \neq j_t}\right) \tag{10}$$

Here the GRU refers to a recurrent unit equal to the one described in the tree encoder in section 2.3.1. Where $\tilde{\mathcal{E}}$ is the edges of a molecular junction tree and i refers to the node traversed at time t.
The model then utilizes a *topological prediction* to binarily predict whether a given node, i_t, still has children to generate. This probability is calculated via a single hidden layer, which is achieved from the latent tree representation, $\mathbf{z}_T$, the

node feature vector $\mathbf{x}_i$ and the inward hidden messages $\mathbf{h}_{k,i_t}$. The expression for the probability can be seen in eq. (11).

$$p_t = \text{Sigmoid}(\mathbf{u}^d \cdot \text{ReLU}[\mathbf{W}_1^d \mathbf{x}_{i_t} + ... \\ ...\mathbf{W}_2^d \mathbf{z}_T + \mathbf{W}_3^d \sum_{(k,i_t)\in\tilde{\mathcal{E}}} \mathbf{h}_{k,i_t}]) \quad (11)$$

Once the probabilities are determined, the child node j, generated from its parent i, will get assigned a node label in accordance with eq. (12).

$$\mathbf{q}_j = \text{softmax}\left(\mathbf{U}_\mathbf{T}^l(\mathbf{W}_1^l \mathbf{z}_T + \mathbf{W}_2^l \mathbf{h}_{ij})\right) \quad (12)$$

Where $\mathbf{q}_j$ is the distribution over the label vocabulary $\mathcal{X}$. It should be noted that when j is a at a root (and thereby has no parent) the inward hidden message, $\mathbf{h}_{ij}$, is 0.
The tree decoder is trained by maximizing the likelihood $p(T|\mathbf{z}_T)$. This is done by minimizing the cross entropy loss functions as seen below, where $\hat{\mathbf{q}}_j$ is the ground truth topological and label values and $\hat{\mathbf{p}}_t \in \{0, 1\}$.

$$\mathcal{L}_c(T) = \sum\nolimits_t \mathcal{L}^d(p_t, \hat{p}_t) + \sum\nolimits_j \mathcal{L}^l(\mathbf{q}_j, \hat{\mathbf{q}}_j) \quad (13)$$

This process follows a teacher forcing method, that replaces the predicted nodes with their ground truth, so that the model predicts based on historically correct predictions, like that of sequence generation.
The prediction is generated without any external guidance, but rather only from recursively guided topological predictions. To make sure that the new tree can be realized into a valid molecule, only clustered labels from set $\mathcal{X}_i$, which are chemically compatible with node i, will be considered when generating child node j. The molecule set $\mathcal{X}_i$ is then sampled over a renormalized distribution, $\mathbf{q}_j$, which is masking out invald labels.

2.3.5. Graph Decoder

The fifth and final component of the model is to reproduce a molecular graph G, from from its predicted junction tree, $\hat{T}$. As many molecules can have the same junction tree, this step is not deterministic. Thus, the degrees of freedom in this problem will vary depending on how neighboring clusters of a molecule are attached to each other. Thus, the goal of the graph decoder is to generate chemically valid molecular graphs. The structure of the graph decoder is given as shown in eq. (14).

$$\hat{G} = \arg \max_{G' \in \mathcal{G}(\hat{T})} f^a(G') \quad (14)$$

Here $\mathcal{G}(\hat{T})$ is a set of graphs which junction tree is given as T, and f^a is a scoring function.
The graph decoder assembles the molecular graph one neighborhood at a time due to this being the most efficient method. The graph decoder begins by assembling the root and its neighbors according to their scores. The assembly then continues outwards.
The realization scoring process of each neighborhood can be explained by having G_i as the resulted subgraph of a merging cluster C_i with its neighbors C_j, where $j \in N_{\hat{T}}(i)$.
G_i is scored as a candidate subgraph by deriving a vector representation $\mathbf{h}_{G_i}$ and using $f_i^a(G_i) = \mathbf{h}_{G_i} \cdot \mathbf{z}_G$ as the scoring function of the subgraph.
The atoms are introduced as u, v in the candidate subgraph G_i where and the indices $a_v = i$ if $v \in C_i$ and $a_v = j$ if $v \in C_j \setminus C_i$. These indices determines the position of the atoms in the junction tree, and allows for retrieval of messages $\hat{\mathbf{m}}_{i,j}$. The messages of the atoms and bonds in subgraph G_i are aggregated into $\mathbf{h}_{G_i}$ in the same way of the encoding step, but with different parameters:

$$\boldsymbol{\mu}_{uv}^{(t)} = \text{ReLU}\left(\mathbf{W}_1^a \mathbf{x}_u + \mathbf{W}_2^a \mathbf{x}_{uv} + \mathbf{W}_3^a \tilde{\boldsymbol{\mu}}_{uv}^{(t-1)}\right) \quad (15)$$

$$\tilde{\boldsymbol{\mu}}_{uv}^{(t-1)} = \begin{cases} \sum_{w\in N(u)\setminus v} \boldsymbol{\mu}_{wu}^{(t-1)} & a_u = a_v \\ \hat{\mathbf{m}}_{a_u,a_v} + \sum_{w\in N(u)\setminus v} \boldsymbol{\mu}_{wu}^{(t-1)} & a_u \neq a_v \end{cases}$$

The model is being augmented with tree messages $\hat{\mathbf{m}}_{a_u,a_v}$ from the tree encoder over the predicted tree $\hat{T}$.

The graph encoder is trained by maximizing the log-likelihood of the predicted correct subgraphs G_i at each true graph G at each tree node as shown in eq. (16).

$$\mathcal{L}_g(G) = \sum_i \left[f^a(G_i) - log \sum_{G_i' \in \mathcal{G}_\rangle} exp(f^a(G_i')) \right] \quad (16)$$

In eq. (16) $\mathcal{G}_i$ indicates the possible subgraph candidate at tree node i. Teacher forcing is used on the graph decoder as well, by having ground truths as input.

2.4. Computational Complexity of the Model

As the JTVAE model consists of the tree decomposer, 2 encoders and 2 decoders, the computational complexity of the model large.
This computational complexity was a large bottleneck throughout this project. The computational complexity yielded a training time of the model on the ZINC dataset of 5 days. Furthermore, training of the model on the QM9 dataset was abandoned after 7 days.

2.5. Similarity Measure

2.5.1. Tanimoto/Jaccard Similarity

To find chemical similarities of two molecules, their molecular fingerprints are compared to each other. These fingerprints consists of a list of bits, where each bit is a chemical property. A common measurement of this similarity is the Tanimoto

coefficient [8]. It measures the common space of the chemical properties, as can bee seen in eq. (17). A low coefficient resembles low similarity, while a high coefficient resembles high similarity between the two molecules.

$$T(A,B) = \frac{A \cap B}{A + B - A \cap B} \tag{17}$$

The RDKit package [6] has built in function for measuring molecules in SMILES format.

2.5.2. *Euclidean Distance in Latent Space*

Another similarity measure suggested by Samanta et al. [9] is a similarity measure based on the euclidean distance in latent space. This similarity measure is shown in eq. (18).

$$EU - Sim(x,y) = 1/\left(1 + \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}\right), \quad x, y \in \mathbb{R}^N \tag{18}$$

This measure is better than the Tanimoto similarity, as it results in molecules with similar chemical properties being close to each other rather than similar molecular structures. It should be noted that a value of 1 corresponds to identical properties (a euclidean distance in latent space of 0), while a value of 0 corresponds to an infinite euclidean distance in latent space.

2.6. Out-of-Distribution Test

When working with generative modelling, it is desired to model a probability distribution of the molecules, $p_\theta(x)$, where θ is the parameters, in which the variational autoencoder. Thus, when sampling a new distribution, we search for a probability close to 1, which indicates identical distributions. It is ideal, that the distributions are close to matching, however they should not be identical, as new molecules with similar chemical features are desired. The Due to i.e. computational limitations, accuracy and simplicity, the log probability is used, as it is a monotone transformation and thereby does not change the properties of the probability. The evidence lower bound (ELBO) was used as an approximation of $\log(p)$. The ELBO can be seen on the right hand side of eq. (19).

$$\log \mathbf{p}_\theta(x) \geq \mathbf{E}_{q(z|x)}\left[\log \mathbf{p}_\theta(x|z)\right] - \mathrm{KL}(\mathbf{q}_\theta(z|x)||p(z)) \tag{19}$$

The log values are used to realize if a given molecule is out-of-distribution. To do so, a threshold is set up, called $\mu_{threshold}$. This parameter is a design choice, and indicates whether a molecule is inside or out-of-distribution. Hence, resulting in eq. (20).

$$\mu_{threshold} > \log \mathbf{p}_\theta(\mathbf{x}_{train}) - \log \mathbf{p}_\theta(\mathbf{x}_{new}) \tag{20}$$

2.7. Novel Similarity Measure

We propose a novel similarity measure, which is a modified euclidean distance. This modified euclidian similarity (MES) combines the euclidian distance with an out-of-distribution test. This measure should be determined as shown in the **Pseudo-Algorithm 1**.

Algorithm 1 Novel similarity measure

Encode molecule, x, using the encoder and obtain $\mathbf{q}_\theta(z|x)$.
Determine if the molecule is inside or out of distribution, $\log p_\theta(x_{train}) - \log p_\theta(x_{new}) < \mu_{threshold}$
if $\log p_\theta(x_{train}) - \log p_\theta(x_{new}) > \mu_{threshold}$ **then**
    Set the modified euclidean distance to 0.
else
    Calculate the euclidean distance as shown in eq. (18)
end if
    return Modified euclidean similarity

We add the out-of-distribution test, as it is still possible for the JTVAE decoder to generate out-of-distribution samples [10, 11], which will result in a poor representation of the given molecule in the latent space. This poor latent representation, may result in the molecule not having the desired molecular features.

3. RESULTS

3.1. Similarity Measure

A molecule from the ZINC dataset was chosen at random and the Tanimoto similarity and the MES measures were calculated between this molecule and all molecules in the ZINC dataset. The SMILES string of the chosen molecule was C[NH+](C/C=C/c1ccco1)CCC(F)(F)F and the chemical structure can be seen in appendix 7.1 Throughout this project it will be refered to as $\mathcal{M}$.
Once the novel similarity measure was calculated, the molecules were then sorted from highest to lowest values with respect to the novel similarity measure. These results can be seen in Fig. 4 (a barplot of the 8 highest scoring molecules and a histogram of the calculated $\log p_\theta(x)$ for each point in the dataset can be found in appendix 7.3 and 7.2).

Looking at the MES compared to the Tanimoto similarity, it is seen that the values of the MES utilizes a much larger range than the Tanimoto similarity. Furthermore, it is seen that some of the molecules achieve a high value of the MES, while a value below 0.2 is achieved of the Tanimoto similarity. This suggests that the MES is able to find molecules, which does not have a chemical similarity, while still having similar latent features. Thus, it seems that the MES are able

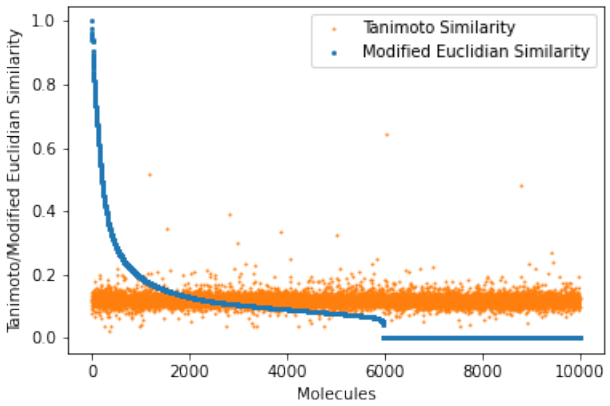


Fig. 4. The modified euclidian similarity and the Tanimoto similarity between molecule $\mathcal{M}$ and all molecules in the ZINC dataset.

to find similarities, which are not apparent from the chemical structures.

3.2. Molecule Generation

Furthermore, the JTVAE decoder was used to generate molecules similar to the randomly selected molecule. The similar molecules were generated by sampling from a Gaussian, $\mathcal{N}(\mu = 1, \sigma = 0.1)$, and adding these to the latent representation of the molecule $\mathbf{z}_{T(\mathcal{M}),G(\mathcal{M})}$. The structure of these new molecules were

4. CONCLUSION

5. ACKNOWLEDGMENTS

We would like to thank Arghya Bhowmik and Peter Bjørn Jørgensen from DTU Energy for supervision, helpful input and discussions throughout the process.

6. SUPPLEMENTARY MATERIALS

The code used in this project is available on GitHub: https://github.com/Junction-Tree-Variational-Auto-Encoder/Novel-Based-Similarity-Measure.git

References

[1] O. Méndez-Lucio, B. Baillif, and D. A. Clevert, "De novo generation of hit-like molecules from gene expression signatures using artificial intelligence," p. 1, 11 2019.

[2] B. Sanchez-Lengeling and A. Aspuru-Guzik, "Inverse molecular design using machine learning: Generative models for matter engineering," p. 3, 07 2018.

[3] United States Environmental Protection Agency, "Smiles," https://archive.epa.gov/med/med_archive_03/web/html/smiles.html, (accessed:27.12.2020).

[4] W. Jin, R. Barzilay, and T. S. Jaakkola, "Junction tree variational autoencoder for molecular graph generation," *CoRR*, vol. abs/1802.04364, 2018.

[5] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato, "Grammar variational autoencoder.," 2017.

[6] G. Landrum, "Rdkit," "https://www.rdkit.org/", (accessed: 02.11.2020)".

[7] Gulcehre C. Cho K. Chung, J. and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling.," 2014.

[8] "Tanimoto/jaccard similarity," "https://www.pilosa.com/use-cases/chemical-similarity-and-the-tanimoto-algorithm/, (accessed: 28.12.2020)".

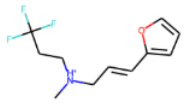
[9] S. Samanta, S. O'Hagan, N. Swainston, T. J. Roberts, and D. B. Kell, "Vae-sim: A novel molecular similarity measure based on a variational autoencoder," 2020.

[10] V. Berger and M. Sebag, "Variational auto-encoder: not all failures are equal," 2020.

[11] Lucas Theis, Aäron van den Oord, and Matthias Bethge, "A note on the evaluation of generative models," 2016.

7. APPENDIX

7.1. Molecule, $\mathcal{M}$



C[NH+](C/C=C/c1ccco1)CCC(F)(F)F

Fig. 5. The molecule with which the similarity measure is calculated.

7.2. Histogram of log p values for all molecules in the ZINC dataset

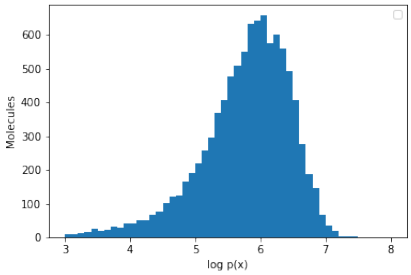


Fig. 6. A histogram of the $\log p_\theta(x)$-values for all molecules in the ZINC dataset

7.3. Modified Euclidian Similarity of the 8 highest scoring molecules.

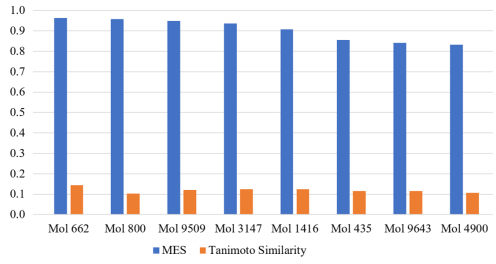


Fig. 7. MES and Tanimoto Similarity for the 8 molecules, which achieves the highest MES score.