Karlsruhe Institute of Technology

Generative models and inverse design

Seminar: Machine learning in the natural sciences

KIT – Karlsruhe Institute of Technology

Tahar Chettaoui

30. march 2022

Principal Supervisor: Jun.-Prof. Dr. Pascal Friederich

Abstract

Identifying new molecules with desirable properties can lead to monumental scientific, technological and biomedical advances. With the tremendous chemical space to explore, computational molecular design is limited by the search strategy and is extremely costly. Recent advances in machine learning produced powerful probabilistic generative models that are able to produce realistic synthetic molecules. Generative models, based on autoencoders (AEs), generative adversarial networks (GANs) or reinforcement learning (RL) have shown promising results. Experiments proofed that the latent space is indeed capturing features relevant to molecules. This leads to the generation of new molecules with desired properties discovered through generative models and inverse molecule design.

Contents

1 Introduction

The drug and material design industry goal is to identify new molecules with desirable properties. The discovery of new molecules can lead to monumental scientific, technological and biomedical advances, but how many potential drug-like molecules are there still to discover? So far we have synthesized about 10^8 substances, whereas the size of the chemical space that is of interest to drug developers is estimated to lie between 10^{23} and 10^{60} [PMV13].

Because of the tremendous molecule space, an important step in molecular design is the generation of a limited number of candidates for computational analysis before experimental synthesis. Virtual screening (VS) is a computational technique used to search the enormous chemical space and filter it to a manageable number of compounds that can be synthesized and tested. Libraries containing thousands to hundreds of millions of candidates can be assessed with simulations or statistical predictions. There are two broad categories of screening techniques: ligand-based and structure-based. The ligand-based drug design relies on knowledge of molecules that bind to the biological target of interest, while the structure-based approach relies on knowledge of the 3D structure of the biological target.

While a discriminative model tries to determine conditional probabilities $p(y|x)$, with y being the molecular property and x a molecular representation. A generative model attempts to determine a joint probability distribution $p(x, y)$, which is the probability of observing both the molecular representation and the physical property. The direct molecular design approach leads from chemical space to the properties, that we can obtain by conditioning the probability on a molecule x to solve $p(y|x)$. On the other hand, the inverse design starts from the desired properties and ends in chemical space. In the second approach, the input is the property and the output is the structure or the distribution of probable structures. This is obtained by conditioning on a property y and by solving $p(x|y)$.

With the number of the currently synthesized substances being negligible compared to what we have still to discover, methods to optimize the identification of new molecules are under development. Machine learning (ML), deep learning (DL), and artificial intelligence (AI) techniques are having huge success in the science industry and its expansion to drug discovery was only natural. Scientists tackled the problem with generative models, such as autoencoders (AEs), generative adversarial networks (GANs) and reinforcement learning (RL).

In this paper, we will go through some of the latest papers presenting solutions based on generative models. In section 2, the methodologies and theory behind each approach will be presented. The first approach will rely on RL, the second one on GANs and the last one on VAE. In section 3, experiments and results of the previously defined methods will be presented before concluding with an outlook in the last section.

2 Methodology

2.1 SMILES chemical notation

The simplified molecular-input line-entry system (SMILES) is a chemical notation that allows a user to represent a chemical structure in a way that can be used by the computer. SMILES strings can be imported by molecule editors for conversion back into two-dimensional drawings or three-dimensional models of the molecules.

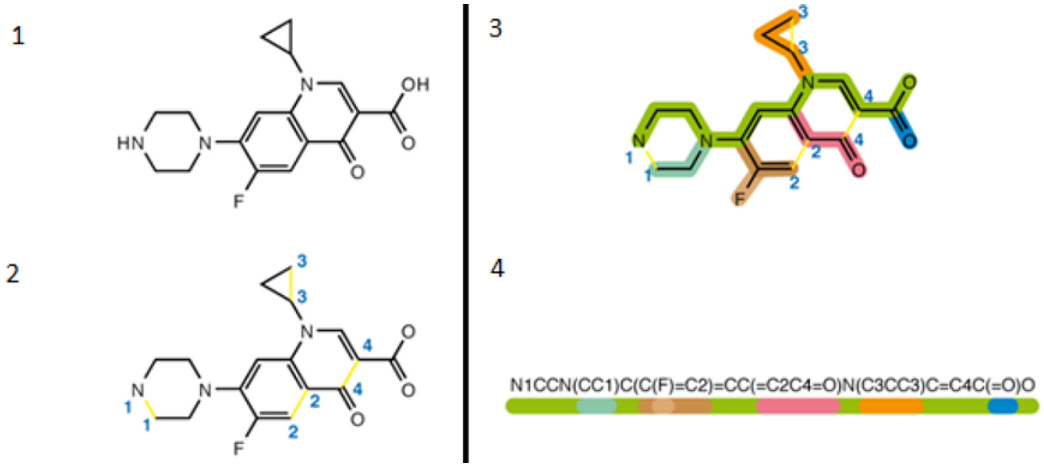

Figure 2.1: SMILES generation algorithm for the Ciprofloxacin molecule [Fda07]

In figure 2.1, the Ciprofloxacin molecule transformation to SMILES notation is illustrated. SMILES is one of many possible chemical representations. Many papers adopted SMILES as input for their models, because this representation can be readily converted into a molecule.

2.2 Approaches

2.2.1 ReLeaSE - RL based

In the RL approach, the agent must learn how to take actions in an environment, which is the world through which the agent lives. Depending on the current state, which is a concrete situation in which the agent finds itself, we assigns positive values to the desired actions in order to encourage the agent to further choose similar actions and we assign negative values to undesired behaviors in order to discourage the agent to follow a wrong path. The goal of the agent is to maximize the cumulative reward.

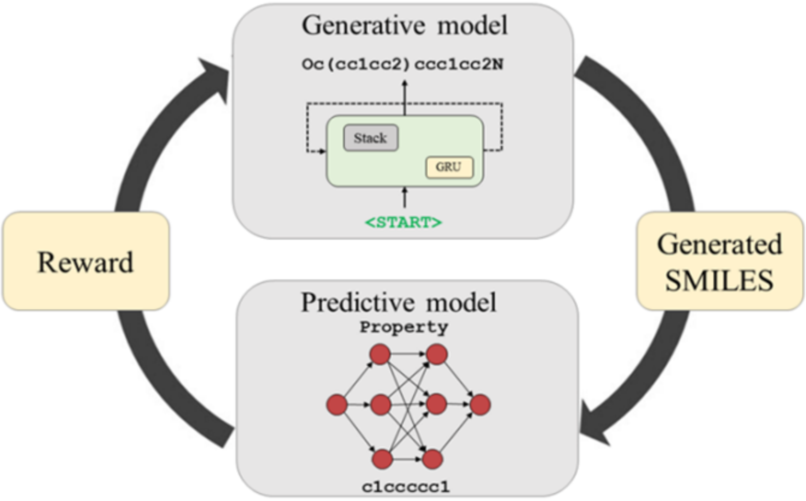

Figure 2.2: ReLeaSE system pipeline [PIT18]

Reinforcement Learning for Structural Evolution (ReLeaSE) [PIT18] integrates two deep neural networks – generative and predictive – that are trained separately but employed jointly. Both models are combined into a single RL system. The generative model is trained to produce chemically feasible SMILES strings, it plays the role of an agent. The predictive model predict the desired properties from a molecule, it plays the role of the environment.

The process of training, illustrated in figure 2.2, consists of two stages: During the first stage, both models are trained separately with supervised learning algorithms. In the second stage, the models are trained jointly with a reinforcement learning approach that optimizes target properties. The predictive model assigns a numerical reward to every generated molecule. The reward is a function of the numerical property generated by the predictive model, and the generative model is trained to maximize the expected reward.

2.2.2 ORGAN - GANs based

The second approach named ORGAN [Gui+17] is based on GANs. GANs are a model architecture for training a generative model. It is composed of a generator and a discriminator. The generator model takes a fixed-length random vector as input and generates a sample in the domain. The discriminator model tries to classify examples as either real or fake, thus leading the generator to learn how to produce realistic examples.

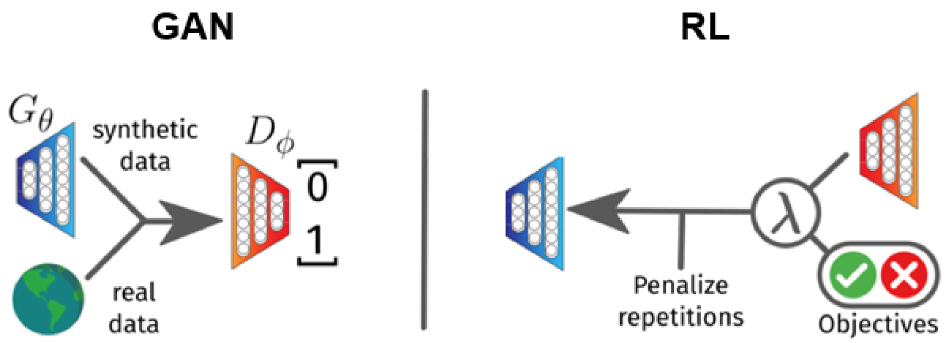

Figure 2.3: ORGAN Architecture [Gui+17]

This work proposes a solution that combines GANs and RL. As presented on the left side of figure 2.3, The discriminator (D) is trained as a classifier by receiving as input a mix of real data and generated data by the generator (G). The discriminator pushes the generator to produce valid molecules' representation. On the right side of the figure, the generator is trained following the principles of RL. The reward or penalty is calculated by combining the discriminator and the objectives. When choosing objectives, they picked qualities that are normally desired, like solubility.

GANs have a number of common failures and none of these problems have been completely solved. GANs might generate samples that are very similar or even identical, because when a generator produces an especially plausible output, the generator may learn to produce only that output. This form of GAN failure is called mode collapse. This work proposes an additional mechanism to prevent mode collapse by diminishing the reward of repeated sequences.

2.2.3 VAE based

In this work [Góm+18], the system takes molecules as input using the SMILES representation. The input is fed to the encoder, which results in a low dimensional vector. In the latent space, a predictor will search for our desired molecule with specific properties. The latent vector with desired properties will then be decoded back to a SMILES representation.

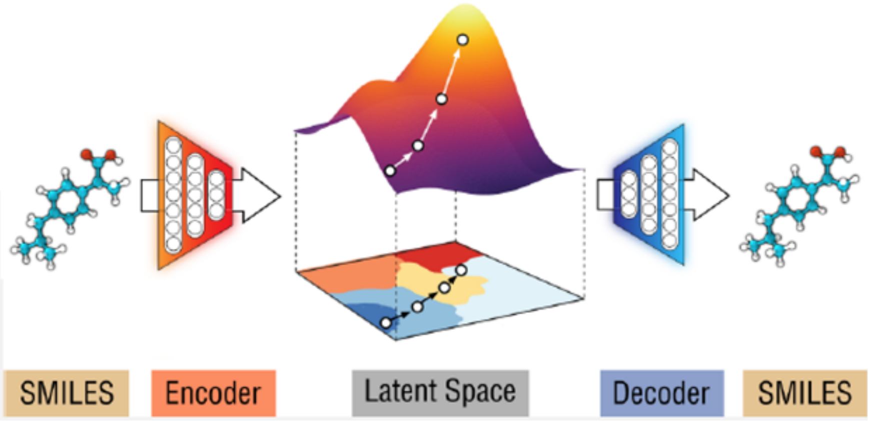

Figure 2.4: VAE jointly-trained on properties [Góm+18]

Autoencoder Vs. Variational autoencoder

Autoencoders (AEs) are neural networks composed of encoders and decoders, as illustrated in figure 2.5. The encoder compresses the input in a lower dimension and produces a vector called the bottleneck. The decoder then reconstructs the input only using this bottleneck. The output is often lossy, since information is lost when reducing the number of dimensions. So the goal is to keep the maximum of information when encoding and the minimum of reconstruction error when decoding. AEs are data-specific, so they can only compress data that is highly similar to data that has already been trained on. The loss function depends on Θ and Φ which are the parameters that define the encoder and the decoder. We are summing up the difference between the original image x, and the reconstructed image $f(g(x))$:

$$L(\theta, \varphi) = \frac{1}{n}\sum_{i=1}^{n}(x^i - f(g(x^i)))^2$$

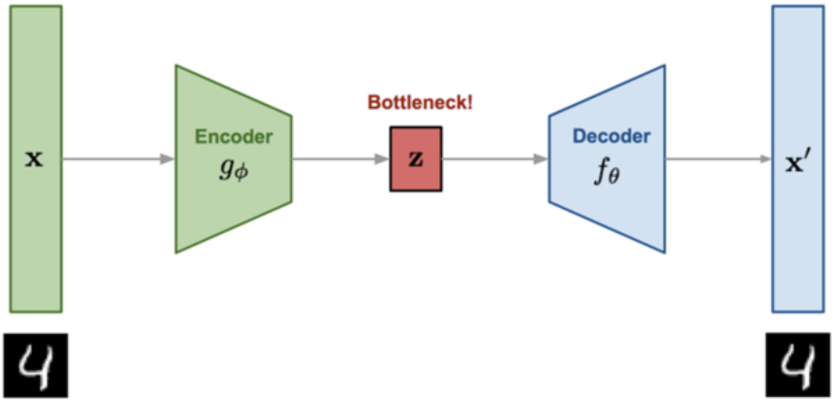

Figure 2.5: Convolutional autoencoder [Ire20]

Unlike AEs, a Variational autoencoder (VAE) learns the parameters of the probability distribution modeling the input data, instead of learning a function. In the figure 2.6, the model learns the mean and standard deviation vector, that are later used to sample a latent vector. For the loss term, we sum up two losses, namely the generative loss and the latent loss. The generative loss is a mean squared error that measures how accurately the network reconstruct the images. The latent loss is the KL divergence, the difference between the encoder's distribution and the latent space distribution.

$$L(\theta, \varphi) = -E_{z \sim q_\theta}\left[P_\theta(x|z)\right] + D_{KL}(q(z|x|)||p(z))$$

$$D_{KL}(P||Q) = \sum_x P(x)log\frac{P(x)}{Q(x)}$$

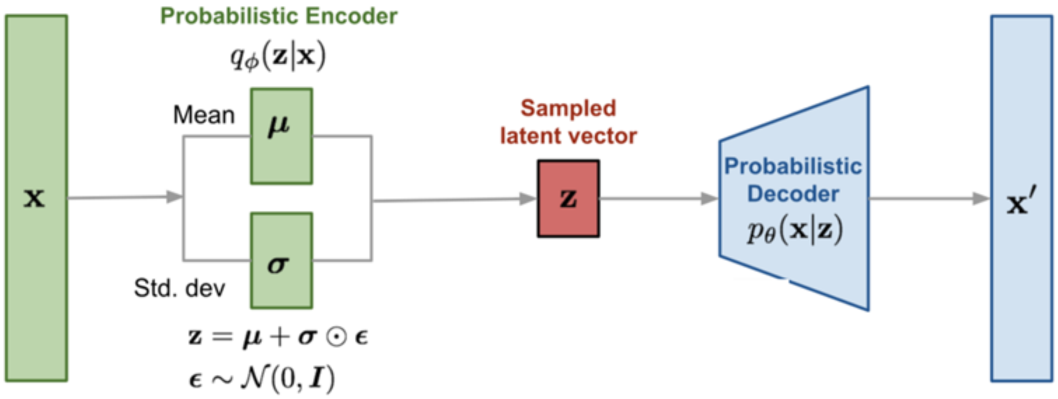
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
$\mathbf{x}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

Figure 2.6: VAE architecture [Ire20]

For unconstrained optimization in the latent space to work, points in the latent space must decode into valid SMILES strings. Without this constraint, the latent space learned by the AEs may contain large "dead areas".This areas decode to invalid SMILES strings. To help ensure that points in the latent space correspond to valid realistic molecules, they chose to use a VAE. The KL loss will push the VAE to learn only valid SMILES representation which solves our problem and allow unconstrained optimization in the latent space. This solution is not perfect, because even with the variational constraint, the character-by-character nature of the SMILES representation can still result in the output of invalid molecules from the decoder.

The predictor

For the training, a deep neural network was trained on chemical structures. First, the encoder converts the discrete representation of a molecule into a real-valued continuous vector. Then, the predictor estimates chemical properties from the latent continuous vector representation of the molecule. The VAE was jointly trained on a property prediction task by adding a multilayer perceptron that predicts property values from the continuous representation generated by the encoder. The regression error of the predictor is included in the loss function. Finally, the decoder converts these continuous vectors back to discrete molecular representations.

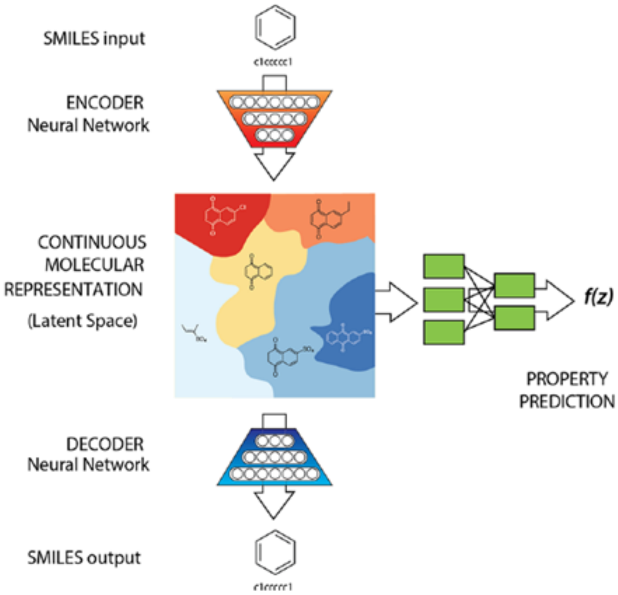

Figure 2.7: The VAE and prediction model pipeline [Góm+18]

To propose a promising new candidate molecules, we can start from the latent vector of an encoded molecule and then move in the direction most likely to improve the desired attribute. The resulting new candidate vectors can then be decoded into corresponding molecules.

3 Experiments

3.1 ReLeaSE - RL base approach evaluationd

In this paper, they defined the reward function as the exponent of the number of benzene rings (−Ph). Figure 3.1 illustrates the evolution of generated structures as the structural reward increases. We see that the model progresses toward generating increasingly more complex, yet realistic, molecules with greater numbers of rings.

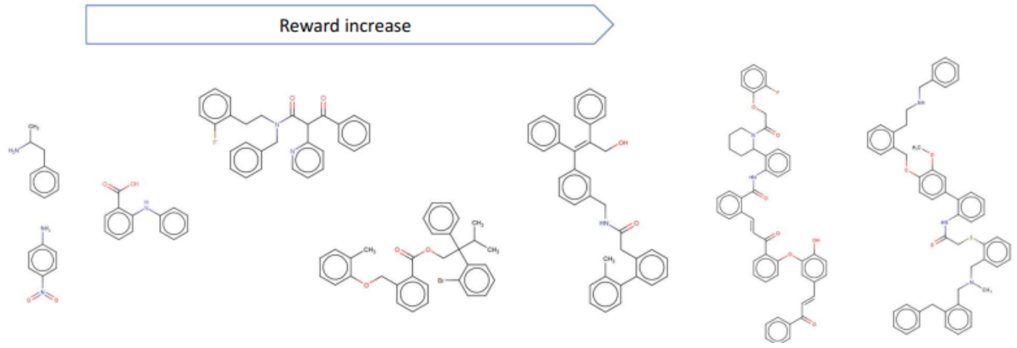

Figure 3.1: Evolution of generated structures as chemical substructure reward increases. Reward proportional to the number of benzene rings [PIT18]

3.2 ORGAN - GANs based approach evaluation

In figure 3.2, we observe Violinplots of Druglikeliness (how likely a molecule is a viable candidate for a drug) for molecules from the baseline Dataset and optimized ORGAN model. Comparing druglikeliness for the data and the best performing approach match showed that ORGAN tends to generate more druglike molecules.

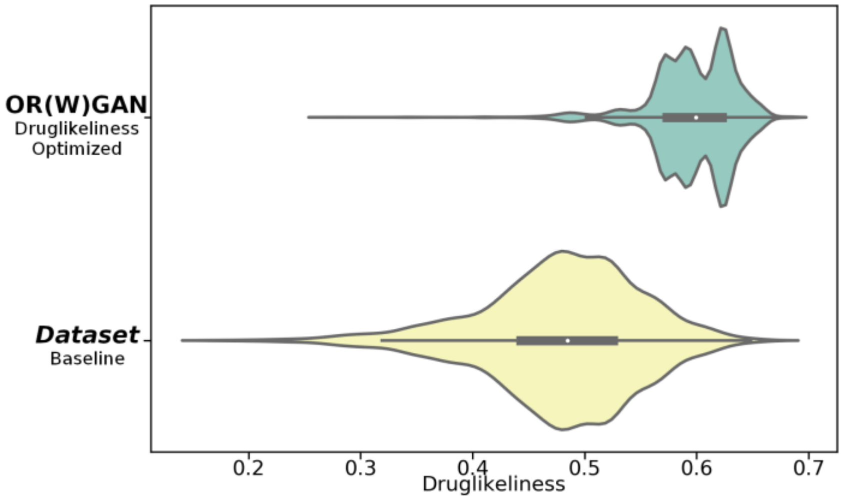

Figure 3.2: Violinplots of Druglikeliness for molecules from the baseline Dataset and optimized ORGAN [Gui+17]

3.3 VAE based approach evaluation

First, they analyze the fidelity of the VAE and the ability of the latent space to capture structural molecular features. the figure 3.3 shows some molecules in the latent space that are close to ibuprofen. These structures become less similar with increasing distance in the latent space. When the distance approaches the average distance of molecules in the training set, the changes are more pronounced, eventually resembling random molecules likely to be sampled from the training set.

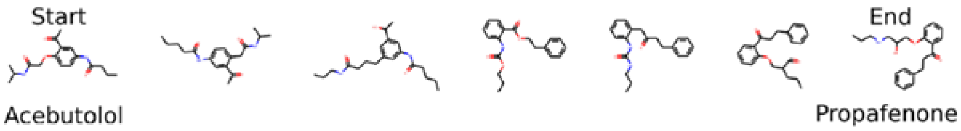

Figure 3.3: Slerp interpolation between two molecules in latent space [Góm+18]

Figure 3.4 shows the probability of decoding the latent representation of a sample drug molecule into several molecules. For most latent points, a main molecule is decoded, and many other slight variations appear with lower frequencies. This is explained by the RNN decoder, since it is stochastic, the decoding might result into multiple molecules. When these resulting SMILES are re-encoded into the latent space, the most frequent decoding also tends to be the one with the lowest Euclidean distance to the original point, indicating the latent space is indeed capturing features relevant to molecules.

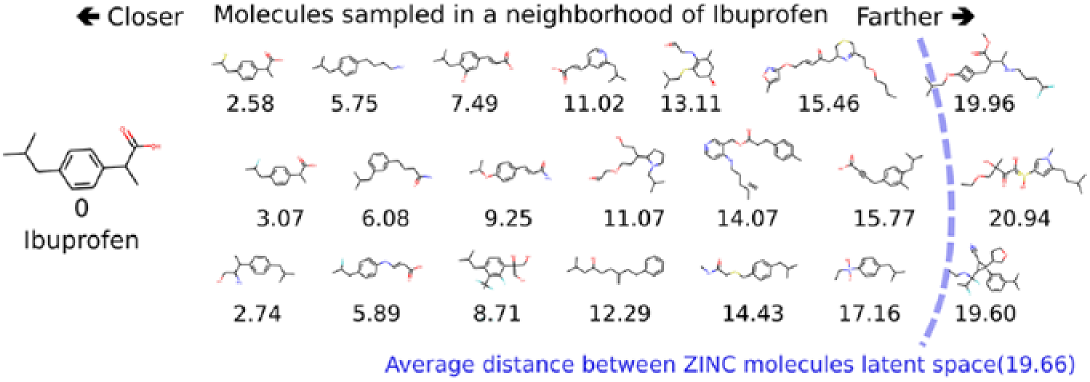

Figure 3.4: Molecules sampled near the location of ibuprofen in latent space [Góm+18]

4 Conclusion

In this paper, three different approaches for inverse molecular design were presented. ReLeaSE was based on RL and integrated two deep neural networks – generative and predictive – that are trained separately but employed jointly. The experiments showed that with an increased reward, the model progresses toward generating increasingly more complex yet realistic molecules. The second approach, namely ORGAN, proposes a solution that combines GANs and RL. The discriminator was trained by receiving as input a mix of real data and generated data by the generator, and the generator is trained by receiving a reward or penalty from the discriminator and the objectives. The third and final approach is based on VAE, which was jointly trained on a property prediction task by adding a multilayer perceptron that predicts property values from the continuous representation generated by the encoder. The experiments showed that the latent space is indeed capturing features relevant to molecules. Since we have so much more to discover in this area and with the exponential growth of the machine learning and deep learning field observed in recent years, these approaches will most likely build the basis for the incoming papers.

During my research on the subject, I came across the Conditional generative adversarial network (CGAN) that could present a viable solution to the problem. In addition to the generator and the discriminator of the GAN, the CGAN have an additional parameter. This parameter is a label given to the latent space to guide the training. An example would be to concatenate the label vector, one-hot encoded, with the bottleneck vector. This label can define the different desired properties of the molecule. The goal here would be to generate the molecule with a desired property from an input molecule and a property as label.

Bibliography

[Fda07] Fdardel. “SMILES generation algorithm for ciprofloxacin: break cycles, then write as branches off a main backbon”. In: *https://en.wikipedia.org/wiki/Simplified*$_{m}olecular-input_{l}ine-entry_{s}ystem/media/File : SMILES.png$ (2007).

[Góm+18] Rafael Gómez-Bombarelli et al. “Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules”. In: *ACS Central Science* 4.2 (2018). PMID: 29532027, pp. 268–276. eprint: https://doi.org/10.1021/acscentsci.7b00572. URL: https://doi.org/10.1021/acscentsci.7b00572.

[Gui+17] Gabriel Lima Guimaraes et al. *Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models.* 2017. URL: https://arxiv.org/abs/1705.10843.

[Ire20] Victor E. Irekponor. “Mathematical Prerequisites For Understanding Autoencoders and Variational Autoencoders (VAEs): Beginner Friendly, Intermediate Exciting, and Expert Refreshing.” In: *https://medium.com/analytics-vidhya/mathematical-prerequisites-for-understanding-autoencoders-and-variational-autoencoders-vaes-8f854025390e* (2020).

[PIT18] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. “Deep reinforcement learning for de novo drug design”. In: *Science Advances* 4.7 (2018), eaap7885. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.aap7885. URL: https://www.science.org/doi/abs/10.1126/sciadv.aap7885.

[PMV13] Pavel G. Polishchuk, Timur I. Madzhidov, and Alexandre Varnek. “Estimation of the size of drug-like chemical space based on GDB-17 data”. In: *Journal of Computer-Aided Molecular Design* 27 (2013), pp. 675–679.

List of Figures