

# Prototype-Based Compound Discovery using Deep Generative Models

Shahar Harel\* and Kira Radinsky\*

*Department of Computer Science, Technion - Israel Institute of Technology*

E-mail: sshahar@cs.technion.ac.il; kirar@cs.technion.ac.il

## Abstract

Designing a new drug is a lengthy and expensive process. As the space of potential molecules is very large,<sup>1</sup> a common technique during drug discovery is to start from a molecule which already has some of the desired properties. An interdisciplinary team of scientists generates hypothesis about the required changes to the prototype. In this work, we develop a deep-learning unsupervised-approach that automatically generates potential drug molecules given a prototype drug. We show that the molecules generated by the system are valid molecules and significantly different from the prototype drug. Out of the compounds generated by the system, we identified 35 known FDA-approved drugs. As an example, our system generated Isoniazid – one of the main drugs for Tuberculosis. We suggest several ranking functions for the generated molecules and present results that the top ten generated molecules per prototype drug contained in our retrospective experiments 23 known FDA-approved drugs.

## Keywords

Prototype-Based Drug Discovery, Compound Design, Generative Models, Deep Learning for Medicine

# 1 Introduction

Producing a new drug is an expensive and lengthy process that might take over 500 million dollars and over 10–15 years. The first stage is drug discovery, in which potential drugs are identified before selecting a candidate drug to progress to clinical trials. Although historically, some drugs have been discovered by accident (e.g., Minoxidil and Penicillin), today more systematic approaches are common. The most common method involves screening large libraries of chemicals in high-throughput screening assays (HTS) to identify an effect on potential targets (usually proteins). The goal of such a process is to identify compounds that might modify the target activity, which might often result in a therapeutic effect.

While HTS is a commonly used method for novel drug discovery, it is common to start from a molecule which already has some of the desired properties. Such a molecule, usually called a “prototype”, might be extracted from a natural product or a drug on the market which could be improved upon. Intuitively, producing a chemically and structurally related substance to an existing active pharmaceutical compound usually improves on the efficacy of the prototype drug – reduces adverse effects, works on patients that are resistant to the prototype, and might be less expensive.<sup>2</sup>

During this process of prototype-based drug discovery, an interdisciplinary team of scientists generates hypothesis about the required changes to the prototype. One might consider this process as a pattern recognition process – chemists, through their work, gain experience in identifying correlations between chemical structure retrosynthetic routes and pharmacological properties.<sup>3</sup> They rely on their expertise and medicinal chemistry intuition to create chemical hypotheses, which have been shown to be biased.<sup>4</sup>

However, the chemical space is virtually infinite – the amount of synthetically valid chemicals which are potentially drug-like molecules is estimated to be between  $10^{23}$  –  $10^{60}$ .<sup>1</sup> In this work, we develop an algorithmic unsupervised approach to automatically generate potential drug molecules given a prototype drug.

It is common to encode molecular structures into SMILES notations (simplified molecular-

input line-entry system) that preserves the chemical structural information. For example, Methyl isocyanate can be encoded using the following string: CN=C=O. We learn to represent drug-like molecules in molecule space represented by SMILES using embeddings. To identify drug-like molecules, which are used to train our algorithm, we use the Lipinski criteria – a common chemical drug-design qualitative measure that estimates the structure bioavailability, solubility in water, potency, etc.<sup>5</sup>

Variational Auto Encoders (VAE)<sup>6</sup> are encoder-decoder architecture that attempts to learn the data distribution in a way that can later be sampled from to generate new examples. State-of-the-art results have been shown for generating images that resemble natural images, yet not identical to the train data.<sup>6,7</sup> Training a vanilla VAE on drug-like molecules provides an ability to sample new molecules which intuitively should be drug-like.<sup>8</sup> In this work, we extend VAE to allow a conditional sampling – sampling an example from the data distribution (drug-like molecules) which is closer to a given input. This allows sampling molecules closer to a prototype drug, and thus increase probability of generating a valid drug with similar characteristics. Additionally, we add a diversity component that allows the sampling to be different from the prototype drug as well. We present a deep-learning approach which we call Prototype-Driven Diversity Networks (*PDDN*), which allows the diverse conditioned sampling. The results show that the molecules *PDDN* generates are similar to the prototype drugs yet significantly diverse. We show empirical results that the system generates high percentage of valid molecules.

Additionally, we perform retrospective experiments and use drugs developed in the 1930’s and 1940’s as prototypes. The system was then able to generate new drugs, some of which discovered dozens of years after the prototype discovery (Figure 1).

One such example is the system discovery of the main drug for Tuberculosis – Isoniazid. Discovered in 1952, it is on the World Health Organization’s List of “Essential Medicines, the most effective and safe medicines needed in a health system”.<sup>9</sup> In the retrospective experiment, we used as prototypes only drugs discovered until 1940. For the drug Pyrazi-

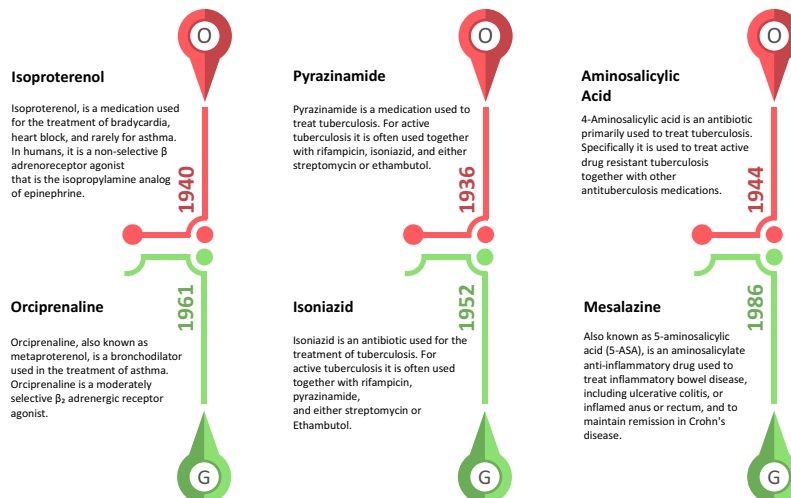


Figure 1: Drug development timeline, with example of drugs generated by *PDDN* (bottom), using FDA approved drugs as prototypes (top).

namide, first discovered in 1936, the system generated the SMILES notation of what today is known as Isoniazid. Pyrazinamide, although discovered in 1936, was not used until 1972 for treating Tuberculosis. Tuberculosis can become resistant to treatment if Pyrazinamide is used alone and therefore is always used in combination with Isoniazid and others. The combination reduces treatment time from 9 months to less than 3 months. This example shows promise on how substances that could not be used at the time of discovery can serve as a prototype for discovering new drugs. We believe our system lays the foundations to build algorithmically-directed HTS based on prototype drugs.

## 2 Related Work

Over the past decade, deep neural networks (DNN) has been a game changer mainly in few areas of machine learning research, such as computer vision,<sup>10,11</sup> natural language processing<sup>12</sup> and speech recognition.<sup>13</sup> More recently, deep learning is rapidly advancing in many more areas of science, such as computational chemistry and pharmacology.<sup>14-18</sup> Deep neural network most prominent success stories are observed in domains with access to large, raw (unprocessed) datasets. In such scenarios deep learning was able to achieve above human

level performance. Compared with those domains, DNN in chemistry relies heavily on engineered features representing a molecule.<sup>19–21</sup> Such approaches are semi-optimal as they restrict the space of potential representations through the assumptions made by limiting to the chosen features.<sup>22</sup>

More recent methods overcome this issue by leveraging advanced deep neural network models to learn chemical continuous representations (i.e., embeddings) based on a large datasets of molecular raw data. Molecular raw data can be represented in few ways, and processed with different deep architectures. Among those we can find 2D/3D images served as input to a convolutional neural network (CNN),<sup>23,24</sup> molecular graph representation paired with neural graph embedding methods,<sup>25,26</sup> and SMILES strings – modeled as a language model with recurrent neural network (RNN).<sup>8,27,28</sup> Others<sup>20,27,29</sup> leverage the embeddings for numerous supervised prediction task, e.g. predicting outcomes of complex organic chemistry reactions.

Recently, deep generative models have opened up new opportunities for leveraging molecular embeddings for unsupervised tasks such as molecule generation and drug discovery.<sup>8,30,31</sup> Most methods aim at generating valid molecules. For example Segler et al. train RNN as a language model to predict the next character in a SMILES string. After training, the model can be used to generate new sequences corresponding to new molecules. Others<sup>8</sup> leverage VAE generative model<sup>6</sup> to learn a dense molecular embedding space. At test time, the model is able to generate new molecules from samples of the prior distribution enforced on the latent representation during training. In this general form of generation, we can only hope to achieve the task of generating molecule libraries with no specific chemical/biological characteristics, but the characteristics of the training data. Others<sup>32</sup> extend this approach by tuning the model on a dataset of molecules with specific characteristics, or by applying post processing methods, such as Bayesian Optimization<sup>8,33</sup> and Reinforcement Learning.<sup>34,35</sup>

In this work, we target the problem of generating drug-like molecules and show that training vanilla generative models on this family shows limited results (Section 5), both for

generating diverse novel molecules and for generating drugs. Following the common chemical approach, we focus the generative approach on a given prototype. This helps “guide” the search process around the prototype in the chemical space. Given prototypes can be drug-like molecules or known drugs. We introduce parametrized diversity and design an end-to-end neural network solution to train the model to represent the chemical space, and to allow for further diversity driven prototype based exploration and novel molecule generation.

### 3 Methods

We define the problem of prototype-driven hypothesis generation as a conditional data generation process. Traditionally, a conditional setting refers to generation from a conditional distribution given some external property. In our case, the model is not conditioned on an external property, but rather, we wish to enable a chemist to provide the model with a known prototype molecule – representing the desired external properties. Thus, we conceptually describe *PDDN* as a conditional molecule generation model although not using pure conditional sampling.

The model operates on a given molecule prototype and generates various molecules as candidates. The generated molecules should be novel and share desired properties with the prototype. The main contribution of our work is enabling prototype-based generation with a diversification factor. We start by reviewing how molecules are represented as text (Section 3.1) and then present a generative model (Section 3.2). Our generative model builds upon recent methods for deep representation learning. We train a stochastic neural network to learn internal molecule representation (embedding). After obtaining the molecule embedding we further utilize the stochastic component of the neural architectures to introduce parametrized diversity layer into the generation process. The architecture of our proposed solution is presented in Section 3.3.

### 3.1 Molecule Representation

The choice of representation of molecules is at the heart of any computer-based chemical analysis. For molecule generation, it is of crucial importance, as the task is to both analyze and generate objects of the same representation. Cadeddu et al. showed that organic molecules contain fragments whose rank distribution is essentially identical to that of sentence fragments. The consequence of this discovery is that the vocabulary of organic chemistry and human language follow very similar laws. Intuitively, there is an analogy between a chemist understanding of a compound and a language speaker understanding of a word. This introduces a potential to leverage recent advances in linguistics-based analysis, and deep sequence models in particular.

A SMILES string is a commonly-used text encoding for organic molecules. SMILES represents a molecule as a sequence of characters corresponding to atoms as well as special characters denoting opening and closure of rings and branches. For example `c` and `C` represent aromatic and aliphatic carbon atoms, `O` represents oxygen, `-`, `=` and `#` represent single, double and triple bonds.<sup>37</sup> Then a molecule, such as Benzene, is represented in SMILES notation as `c1ccccc1`. It has already been shown that SMILES representation of molecules has been effective in chemoinformatics.<sup>8,27,28,32</sup> This has strengthened our belief that recent advances in the field of deep computational linguistics and generative models might have an immense impact on prototype based drug development.

### 3.2 Molecule Driven Hypothesis Generation

Generative models have been applied for many tasks, e.g., image generation. The models synthesized new images which resembled the database the models were trained on.<sup>6,7</sup> One of the most popular generative frameworks are Variational Autoencoders (VAE).<sup>6</sup>

VAE are encoder-decoder models that use a variational approach for latent representation learning. Intuitively, the encoder is constrained to generate latent representations that follow a prior. During generation, latent vectors are sampled from the priors and passed

to the decoder that generates the new representation. We leverage VAE for the task of molecule generation. The stochasticity allows integrating chemical diversity into the generation process. However, application of generative models for molecule generation have shown limited results.<sup>8</sup> Unlike image generation, where each image is valid, when we aim at molecule generation, not each representation is a valid molecule representation. Intuitively, when we sample from the prior for image generation – the space of images is much more dense than that of valid molecules. Therefore, many image samples are valid compared to randomly generated molecules representations. We hypothesize that a constrained generation next to a known prototype, rather than a non-constrained sampling, will yield better molecule generation. We extend VAE generation process to condition on a prototype, i.e., generate molecules closer to a given drug. Intuitively, directing the sampling process closer to existing prototype drugs might yield valid molecules that carry similar characteristics to the the prototype yet provide diversity. Our results provide evidence that a conditioned sample along side a diversity component yields more valid and novel results. If conditioned on known drugs, the system is able to generate drugs discovered years after the prototype (Section 3.2).

More formally, we assume a molecule  $M$  has a latent representation  $z$  that captures the main factors of variation in the chemical space. We model the covariates  $z_i | M$  as Gaussians ( $z_i \sim \mathcal{N}(\mu_i, \sigma_i)$ ). With the latent representation  $z$  at hand, we want to generate a candidate molecule in a SMILES discrete form, therefore we define the generative model  $\hat{y} | z \sim Multi(\theta)$ , where  $y$  is the generated candidate, Formally:

$$q(z | M) = \prod_{i=1}^{D_z} \mathcal{N}(\mu_i = \hat{\mu}_i, \sigma_i = \hat{\sigma}_i) \quad (1)$$

$$p(\hat{Y} | z) = Multi(\theta = \hat{\theta}) \quad (2)$$

Where  $q$  is approximated via encoder neural network function, applied on molecule  $M$  as input, and outputs the latent feature parameterization  $(\hat{\mu}, \hat{\sigma})$  of the molecule. We than



sample instance from this parametrization to obtain the final encoded output  $z$ ;  $p$  is represented via decoder neural network, applied on the molecule sampled feature instance  $z$  as input, and generates the output molecule as described below.

Generating a molecule as a SMILES string reflects multinomial distribution over the atoms space. Each atom is represented via a character. We form the character generation process as an iterative process, each character  $y_i$  is generated based the the hidden encoded representation  $z$  and the formerly generated  $y_i$ 's. In total, the output of this step is a string  $\hat{y}_i = \{\hat{y}_1, \dots, \hat{y}_N\}$ , where  $N$  is a pre-defined maximal generation length. Formally, for a single character  $y_i$

$$P(\hat{y}_i | \hat{y}_1, \dots, y_{i-1}, z) = f(s_i, y_{i-1}) \quad (3)$$

Where  $y_{i-1}$  is the character embedding correspond to the last character generated, and  $s_i$  is a state at step  $i$  representing the current processed information of both the molecule latent representation  $z$ , and the formerly generated  $y_i$ 's up to  $i-2$ .

Our goal is to create a molecule that is different from the original molecule  $M$ . Intuitively, we wish to explore the chemical space around the molecule  $M$ . Therefore, during generation process we introduce a diversity component noising the multidimensional Gaussian parameters used for sampling the hidden vector  $z$ . More formally, to introduce *diversity* to our generation process, we instantiate our encoder output parameters with a diversity layer. Intuitively, the diversity layer outputs a noisy sample from a distribution centered as the encoder suggested, but with larger variance. This allows us to explore the molecule space around an origin molecule, with tune-able amount of diversity, corresponds to variability in chemical space. The diversity layer samples noisy instance according to the encoded Gaussian parameters and a diversity parameter  $D$ .

The output of the diversity layer is a sample from a conditional diverse distribution, described as follows:

Given the encoder outputs: vector of means  $\hat{\mu}_i$  and standard deviations  $\hat{\sigma}_i$ , and random

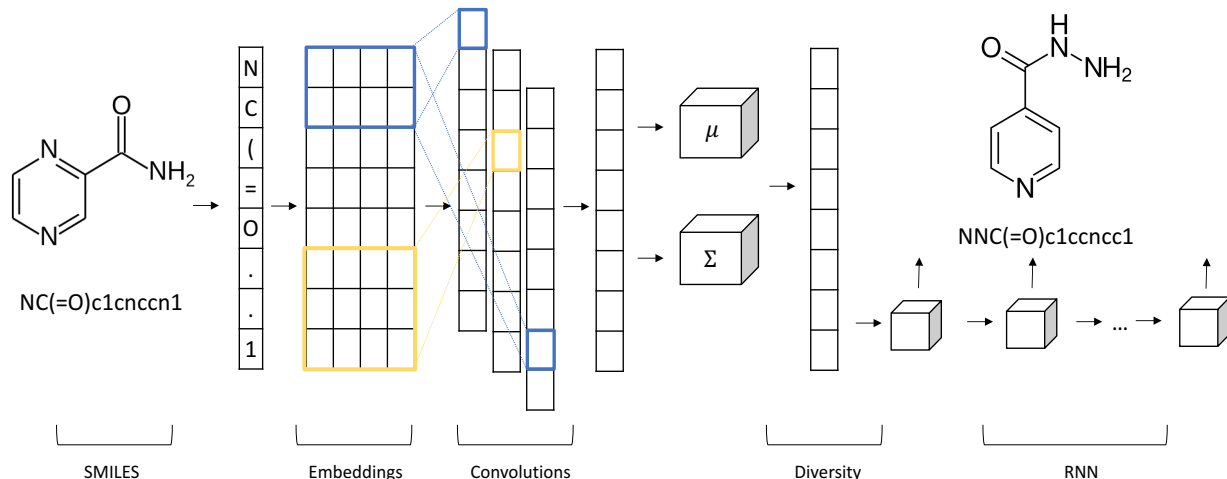


Figure 2: *PDDN* end-to-end neural net architecture

noise sample  $n$  - from Gaussian distribution with diversity parameter  $D$  - [ $n \sim \mathcal{N}(O, D)$ ].

$$\text{Diverse } z = (n \times \hat{\sigma}_i) + \hat{\mu}_i \sim \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2 \times D) \quad (4)$$

We obtain instance from the diverse distribution as our final noisy encoded representation ( $z$ ) for the compound  $M$ , used as the base to for the decoder diversity-driven molecule generation.

We note that during training our diversity parameter  $D$  is set to 1. Thus  $z$  instance is sampled from the non-diverse distribution suggested by the computed parameters. Tuning this parameter at generation time allow us to explore the space around the prototype.

### 3.3 *PDDN* Architecture

We leverage recent advances in generative models and deep learning for natural language process (NLP) to form the prototype hypothesis generation process as an end-to-end deep neural network solution. Figure 2 presents *PDDN* (Prototype-Driven Diversity Network) architecture.

*PDDN* starts by encoding the molecule (in SMILES notation) using the encoder function. First, encoding each character in the SMILES representation into its  $d$  dimensional

embedding, then applying convolutions over various substring (filter) sizes (e.g. correspond to chemical substructures). A similar encoder architecture was suggested for NLP tasks, such as sentence classification.<sup>38</sup> The extracted features are then concatenated and fully connected layers are applied. The outputs of the encoder are considered as a vector of means and a vector of standard deviations, representing the distributions of features for the prototype. In VAE, those vectors are then fed into a decoder. The goal is to optimize reconstruction of the original input and constraint the representation to a known prior. During generation, the vectors of features are sampled from the prior distribution and their output is passed to the decoder that generates a new representation.

We extend the VAE generation process by adding a *diversity layer*. During generation, instead of sampling from the prior means and standard deviations, we first feed a prototype. We sample from the prototype feature distribution with parametrized diversity (Section 3.2) to form the prototype latent representation – served as input for the decoder.

As described in section 3.2, our decoder is a sequential generator. By generating sequentially, we form another parameter of variability in the generated data, by introducing minor variations into the molecule generated during generation. This is the main component of many other works on molecule generation<sup>32,39,40</sup> to introduce diversity into the generation process. We later show that our diversity layer can introduce diversity beyond this component.

We represent our decoder as a recurrent neural network (LSTM<sup>41</sup>). The decoder receives the encoder output as its input. The encoded representation forms the first state of the decoder. The decoder then generates the compound sequentially (character by character) by operating on the distribution over characters at each time step, based on its updated state and the input character from former step.

During training, we feed the decoder with the correct next symbol, even if it was predicted wrongly.<sup>42</sup> During generation, we experiment with two options for generating the next symbol: one by selecting the best scored character from distribution over symbols (argmax),

and the second is by sampling from the same distribution. By introducing sampling into the generation, we are able to increase the amount of variability we generate during generation. The model is trained to reconstruct the input prototype from a low dimensional continuous representation, by minimizing a discrete reconstruction loss. Formally, to minimize the reconstruction error on a discrete molecule representation, we use the cross-entropy loss, defined as:

$$H(y, \hat{y}) = \sum y_i \log \hat{y}_i \quad (5)$$

We note that we minimize the variational lower bound,<sup>6</sup> which is essentially optimizing the reconstruction error while constraining the latent distribution with a prior. To reconstruct syntactically valid SMILES, the generative model would have to learn the SMILES, which includes keeping track of atoms, rings and brackets to eventually close them. In this case, the lower dimension representation that can be reconstructed into a valid prototype, is a highly informative representation. In other words, by minimizing the reconstruction error, we want learn a prototype continuous representation that captures the coordinates along the main factors of variation in the chemical space. This representation is the base for further diversifying the molecule generation process.

### 3.4 Ranking the Generated Population

In previous sections, we presented a method for generating molecules given a prototype. To better focus a drug-design process, we would like to be able to present a ranked-list of potential drug candidates. As a compound development is a long and expensive process, such ranking is of high importance. In many cases, examining hundreds of compounds might be intractable.

When ranking, one should optimize for an objective, e.g, aiming at specific molecular characteristics (logP, Molar refractivity etc.). In this work, we develop two methods for

ranking the generated population with respect to a prototype input.

1. **Lipinski-based Ranking:** We rank the generated molecules in proportion to the number of violations to Lipinski’s rule of five.<sup>5</sup> This rule is a heuristic that evaluates the druglikeness of a molecule. It verifies several chemical properties and physical properties that would make it a likely orally active drug in humans. Intuitively, the prototype drug represents the characteristics or targets one would like the generated drug to poses. During our ranking, we therefore, prefer candidates that carry resemblance to the prototype drug: we represent each molecule via a finger-print vector representations<sup>43</sup> and use as a secondary ranking criteria, the Tanimoto coefficient to calculate molecular similarity between the generated compounds and the prototype (i.e., ranking higher a generated molecule with the least violations to Lipinski rule, and the most similar to the prototype).
2. **Target-based Ranking:** It is common<sup>44,45</sup> to evaluate a potential drug candidate by predicting its binding profile – the possibility of being active on a desired biological target. A common method for this purpose is to identify similar compounds, whose target binding profiles are known, and infer that the same targets might be relevant for the potential drug candidate.<sup>43</sup> Following this observation, and our goal of generating molecules with similar characteristics to a prototype, we build a ranking mechanism that emphasizes the prototype target binding profile. We used priorly trained ligand-based target prediction model<sup>46</sup> obtained from ChEMBL<sup>47</sup> to obtain probability vector over approximately 1,500 targets for both the prototype and each of the generated compounds. Finally, we rank the generated molecules in proportion to the cosine similarity with the input prototype – represented by the target probabilities vector.

Table 1: *PDDN* hyperparameter configuration

Parameter	Value
max molecule length	50
char embedding size	128
filter sizes	3, 4, 5, 6
number of filters	128
latent z dimension	300
batch size	64
initial learning rate	0.001
LSTM cell units	150

## 4 Experimental Settings

In this section we provide details on the datasets, hyperparameter setting, and the training in general. Then, we mention the methods compared and used in our experiments.

### 4.1 Model Details

*PDDN* was trained using a Tensorflow API.<sup>48</sup> We use the Adam algorithm<sup>49</sup> to optimize all the parameters of the network jointly, regarding weights initialization - the atoms embedding were initialized using a random uniform distribution ranging from -0.1 to 0.1, convolution weights used truncated normal with std 0.1, all other weights used the Xavier initialization,<sup>50</sup> biases were initialized with constant. To reduce overfitting, we included an early stopping criteria based on the validation set reconstruction error. We use exponential decay factor on learning rate, and the teacher forcing method<sup>42</sup> during training. In total, table 1 presents *PDDN* hyper parameter configuration.

The code for our system is available over github<sup>1</sup> for further research in the community

---

<sup>1</sup>[https://github.com/shaharharel/CDN\\_Molecule](https://github.com/shaharharel/CDN_Molecule)

## 4.2 Datasets

### 4.2.1 Drug-like molecules database

In our work, we provide experiments showing that CDN is capable of generating drug-like molecules. We train our model on a large drug-like molecules database and present several metrics on the generated molecules. The ZINC database<sup>51</sup> contains commercially available compounds for structure based virtual screening. In addition, the database have subsets of ZINC filtered by physical properties. One such filtering is based on Lipinski’s rule of five<sup>5</sup> – a heuristic method to evaluate if a molecule can be a drug. The subset contains over 10 million unique drug-like compounds. *PDDN* was trained on a subset with approximately 200k drug-like compounds extracted at random from the ZINC drug-like database. The subset was further divided to train/validation/test sets, with 5k compounds for validation and test sets, and the rest for training set. The subsets are used for training the model (train), evaluating hyperparameters and stopping criteria (validation), and for method evaluation and experiments (test).

### 4.2.2 Drug database

For our drug-generation experiment (Section 5.2) we show that some of the molecules generated by CDN are drugs which were discovered years later. The DrugBank database<sup>52</sup> is a bioinformatics and cheminformatics resource that combines detailed drug data with comprehensive drug target information. For retrospective experiments, we extracted a test set of 869 FDA approved drugs from the DrugBank database. Note, our system is not trained on drugs, but rather presented with drug prototypes only during generation.

## 4.3 Compared Methods

As discussed in Section 2, not much work has been done in the area of deep drug generation and specifically not on the conditional setting and the diversity aspect of the generation.

To the best of our knowledge, the works that do target a similar task, train for unconditional molecule generation and later apply post-processing to achieve general molecular characteristics. For example, Olivecrona et al.<sup>35</sup> apply reinforcement learning over a generated molecule to resemble another predefined molecule. We compare our methods to the state of the art models for molecule generation on the reconstruction criteria, and further show that our model is able to build on top of those models to apply diversity. Specifically, we compare all following methods

1. *Seq2Seq*<sup>53</sup> - An autoencoder architecture applied on sequence data for prediction of sequences. Both encoder and decoder are recurrent neural networks (RNN). Although the model is in general deterministic, it is able to bring stochasticity (and thus novelty) into the molecules generation process by setting the RNN decoder to sample from the distribution over characters in every time step instead of predicting the topmost next character. We therefore consider two baselines – one using the Argmax method and the other utilizing the Sampling method to reach diversity.
2. *Conv2Seq* - To conform better with *PDDN* parameter setting that utilize on CNNs, we implement a second auto encoder same as the previous method but with convolution encoder.
3. VAE<sup>6</sup>– a vanilla implementation of VAE. This model generates new molecules from unit Gaussian random samples, regardless of prototypes.
4. *PDDN* -VAE - Our diversity model on top of variational auto encoder.  $D$  is the diversity parameter of Equation 4 The higher the  $D$ , the higher the diversity induced. We note, that for  $D = 1$ , the model extends VAE for a conditional setting but without diversity.



## 5 Experiments

In this section, we first conduct several experiments to determine *PDDN* performance in the task of reconstructing the molecular structure. We present evaluation of the trade off between the molecules reconstruction accuracy and novelty as a function of *PDDN* diversity component. Additionally, we conduct several drug related experiments to show *PDDN* capabilities in the real world for generating new drugs.

### 5.1 Novel Molecules Generation

Our main goal is to create novel molecules that carry similarities to the prototype. Thus, the metric of reconstruction is an important metric. We examine the methods on the task of prototype reconstruction on a test set of 5k ZINC drug-like compounds. To explicitly address the reconstruction accuracy and validity vs. the generated molecules diversity, we measure the following metrics:

1. *Reconstruction Accuracy (Acc)* - Character-level accuracy with the input prototype served also as target.
2. *Valid Molecule Percentage (Valid)* - Percentage of valid molecules. There are several numeric validations performed on molecules representation to validate its correctness. We used Rdkit<sup>54</sup> library to measure validity of the generated compounds.
3. *Novel Molecule Percentage (Novel)* - A novel molecule is both a valid molecule, and different from the prototype.

To be able to measure the molecule generation capabilities over various Gaussian samples for the same prototype compound (we want to be able to generate *several* compounds related to the origin compound), we also measure all the above metrics with @*k* notation. In our context, @*k* represents that for each prototype compound, we run *PDDN* generation process with *k* instances of random noises parametrized with diversity *D*. We note that to measure

Table 2: Evaluation of *PDDN* and baselines for diversity and validity of generated molecules

Model	Acc	Valid	Novel	Acc @ 1k	Valid @ 1k	Novel @ 1k
Seq2Seq - Argmax	0.94	0.93	0.13	-	-	-
Seq2Seq - Sampling	0.91	0.88	0.19	0.92	0.89	32.5
Conv2Seq - Argmax	0.92	0.85	0.14	-	-	-
Conv2Seq - Sampling	0.89	0.77	0.18	0.88	0.76	35.2
VAE	<sup>2</sup>	0.58	-	-	-	-
<i>PDDN</i> - D=1	0.91	0.89	0.19	0.9	0.89	8
<i>PDDN</i> - D=2	0.82	0.81	<b>0.26</b>	0.81	0.8	<b>66.6</b>
<i>PDDN</i> - D=3	0.64	0.63	<b>0.37</b>	0.65	0.65	<b>227</b>

*Novel@k*, we count how many unique molecules generated – a novel molecule is counted only once, even if it was generated with various Gaussian samples for the prototype. The metric of *Novel@k* is not normalized, thus, the semantics of this metric should be, intuitively, interpreted as how many unique molecules were generated for a prototype and 1000 Gaussian samples.

Table 2 presents the results of *PDDN* and the baselines on the metrics above. Analyzing *Acc*, *Valid* and *Novel* metrics, we observe that with diversity level of 1 (non-diverse sampling), *PDDN* generates similar diversity to the baselines. Increasing *D*, significantly increases the diversity in the generated molecules, while reducing the level of accuracy and the valid molecules rate. This result stems from the intuition that as the representation becomes noisier, it is harder for the model to reconstruct the original prototype. Addressing the @*k* metrics, we observe *PDDN* is able to maintain the accuracy and validity levels with many random samples used for generation, while generating various unique molecules for the same prototype input, with the number of unique molecule significantly increasing with the diversity parameter *D*.

## 5.2 Drug Generation

The main aim of this work is to generate novel molecules with desired properties (characterized by the prototype molecule), by searching the chemical space around the prototype.

<sup>2</sup>As no prototype is given, there is no reconstruction to measure

Table 3: Sample of automatically generated drugs and the drug served as prototype to the generation process

Input Drug	Input SMILES	Generated Drug	Generated SMILES
Aminosalicylic	<chem>Nc1ccc(C(=O)O)c(O)c1</chem>	Mesalazine	<chem>Nc1ccc(O)c(C(=O)O)c1</chem>
Pyrazinamide	<chem>NC(=O)c1cnccn1</chem>	Isoniazid	<chem>NNC(=O)c1ccncc1</chem>
Protriptyline	<chem>CNCCCC1c2ccccc2C=Cc2ccccc21</chem>	Desipramine	<chem>CNCCCCN1c2ccccc2CCc2ccccc21</chem>
Phenelzine	<chem>NNCCc1ccccc1</chem>	Isoniazid	<chem>NNC(=O)c1ccncc1</chem>
Isoproterenol	<chem>CC(C)NCC(O)c1ccc(O)c(O)c1</chem>	Orciprenaline	<chem>CC(C)NCC(O)c1cc(O)cc(O)c1</chem>
Pheniramine	<chem>CN(C)CCC(c1ccccc1)c1cccn1</chem>	Tripeleennamine	<chem>CN(C)CCN(Cc1ccccc1)c1cccn1</chem>

### 5.2.1 Re-discovering Known Drugs

To check the immediate benefits (i.e., without further screening the generated compound) of our approach to a real world task, we conduct a retrospective experiment in the drug domain. We apply our method on a test set of FDA approved drugs as prototypes. We note that none of the drugs was observed in the training data, which was composed of only drug-like molecules.

Evaluation on this task is harder since our goal is to generate drugs, and we cannot a-priori know if the generated molecule has the desired characteristics of a drug without further experimenting with the compound. We therefore consider as gold standard a test set of 869 approved known drugs. Although this test set is very small in compare with the enormous molecule space, some approved drugs are chemically similar and share similar therapeutic characteristics, thus we hypothesize that by applying *PDDN* on FDA approved drugs as prototypes, we might be able to generate other known compounds / drugs with similar characteristics.

In this evaluation, we target existing drugs as prototypes – by feeding them one by one in a leave-one-out cross validation manner to *PDDN*. The generated molecules are compared with the rest of the FDA approved drug list. Interestingly, our model was able to generate molecules that also appear in the FDA approved drugs list and are closely related to the prototype, both in the chemical aspects and by their medical use (i.e. targeting the same biological mechanism of action). Table 3 presents a sample of the drugs generated.

Table 4: Automatically generated FDA approved drugs. We present the percentage of the FDA-approved drug from the total valid molecules generated by each method

Model	#Drugs	% from Generated Valid Molecules
VAE	0	0%
Seq2Seq	12	0.002%
Conv2Seq	9	0.0018%
CDN-VAE D=1	12	0.0023%
CDN-VAE D=2	22	0.005%
CDN-VAE D=3	35	<b>0.01%</b>

### 5.2.2 Retrospective Generation Experiments

We run the baselines and *PDDN* variants over all 869 approved drugs dataset as prototypes, with 1000 Gaussian samples in each run. Table 4 presents the total number of FDA approved generated drugs with each method (with no ranking – evaluating all the generated compounds). We also present the percentage those drugs constitute from the valid molecules generated. The results seem promising, especially given the negligible chance of generating a drug using exhaustive search without constraints (e.g., using HTS). We observe that the VAE could not produce any known drug. We hypothesize that this stems from the fact that VAE randomly generates a molecule and not based on a prototype. *PDDN* with no diversity and the other baselines generated 9–12 unique drugs. This result emphasize how the variability that the decoders present during sampling contributes to the generation of known drugs. More interestingly, we observe that for higher  $D$  values of our diversity layer (CDN-VAE D=2 and CDN-VAE D=3), the amount of unique known drugs increases significantly.

We observe that larger diversity rates generated significantly more medical discoveries than other methods (in form of generating FDA approved drugs), although one would expect that larger diversity might generate noisier examples and thus less medical discoveries. Our diversity layer results in samples that do not coincide with the prior distribution enforced during training over the latent space (Section 3.2), unlike VAE and *PDDN* with  $D = 1$ . Thus, such sampling results in final marginal distribution that does not match the prior, which in turn results in lower number of valid molecules. However, we are more interested in novel

Table 5: Precision@10/100 of automatically generated FDA approved drugs using *PDDN* for generating and ranking the molecules. L stands for Lipinski 5 based ranking and TP for target prediction based ranking.

Model	#Drugs@10 - L	#Drugs@100 - L	#Drugs@10 - TP	#Drugs@100 - TP
VAE	0	0	0	0
Seq2Seq	10	12	10	12
Conv2Seq	7	9	8	9
CDN-VAE D=1	11	12	10	12
CDN-VAE D=2	19	22	19	22
CDN-VAE D=3	23	31	17	31

molecules and medical discoveries, even on the expense of valid molecules. Thus, with the idea of exploring the latent space, we add the diversity layer on top of the prior. In Table 3, we observe that methods that utilize larger diversity, and thus sample from the “diversified” marginal distribution that does not match the prior, result in medical discoveries that were not observed in methods that did follow prior. For example, both *PDDN* with  $D = 1$  and vanilla VAE are sampling according to the prior and are able to generate large rates of valid molecules, while *PDDN* with  $D = 3$  which samples from the diversified distribution generated less valid molecules but significantly larger rates of novel molecules which resulted in more FDA approved drugs.

### 5.2.3 Retrospective Candidate Ranking Experiments

Table 5 presents generation and ranking performance of *PDDN* with both ranking methods (see Section 3.4). We present the precision@10 and precision@100. In other words, the number of FDA approved drugs found within the top 10 or top 100 ranked generated molecules.

We observe both ranking methods are able to capture most of the generated drugs within its top ranked candidates. We observe that *PDDN* generates known drugs within the top 10 ranked as well as within the top 100. This shows promise about the quality of such generated molecules. We observe an additional phenomena – although the higher the diversity the higher the amount of novel generated molecules (Table 2), the precision@10 and pre-

recision@100 of those methods is higher. This indicates that the diversity methods not only generated more novel molecules, but also had a higher precision at the top ranked molecules.

One should remember that the model doesn't have any "drug" understanding – the model was only trained given drug-like molecules, and *all known drugs were eliminated from the training*. The key here is the chemical similarity drugs share. Thus, by targeting a drug molecule as prototype to the generation process, our model is able to chemically diversify the prototype drug in a way that generated another known drugs. We are encouraged by the results that *PDDN* was able to generate a significant number of previously known drugs during the retrospective experiment.

### 5.3 Qualitative Examples

We present a few qualitative examples of the drugs generated. We would like to explore whether the application of the system on drugs developed up until a certain year might find drugs that will be discovered years later. During training we eliminate all known drugs from the ZINC database and we present as prototypes a single drug. Figure 1 presents a timeline with example pairs of origin (top row) and generated (bottom row) molecules, with the year of the drug first use. By using *PDDN* we could have generated the bottom molecules directly when we knew the origin molecules, possibly sparing years of research. The system was able to identify the main drug for Tuberculosis – Isoniazid using an initial prototype of the disease that was never used due to its side effects (Pyrazinamide). Additional intriguing example is the generation of Orciprenaline which is used to treat Asthma from a prototype drug that was mainly used for heart block, and very rarely for asthma. These pairs are closely related in their therapeutic effect, but a few changes for the molecule were needed to reposition it for Asthma treatment. Another interesting discovery was Mesalazine, used to treat inflammatory bowel disease based on an antibiotic primarily used to treat tuberculosis. discovered about 40 years before.

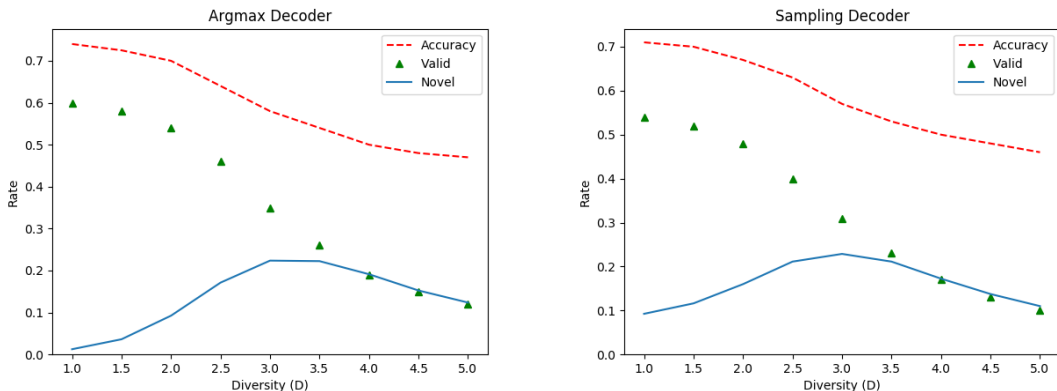


Figure 3: Diversity parameter effect on performance.

Table 6: *PDDN* performance using a sampling decoder.

Model	Acc	Valid	Novel	A@1k	V@1k	N@1k
<i>PDDN</i> D=1	0.88	0.78	0.19	0.88	0.79	39.5
<i>PDDN</i> D=2	0.8	0.69	<b>0.25</b>	0.79	0.68	<b>94</b>
<i>PDDN</i> D=3	0.57	0.39	<b>0.27</b>	0.56	0.38	<b>179</b>

## 5.4 Diversity Mechanisms

A common method to employ diversity in encoder-decoder models is to employ a sampling decoder into the architecture.<sup>32,39,40</sup> The diversity is introduced by sampling from distribution over characters in each time step of generation, rather than choosing the topmost (argmax) character at test time. We analyze the contribution of the diversity layer  $D$  for *PDDN* presented in this work along side a sampling decoder as well. Table 6 presents *PDDN* performance on the previous metrics (Section 5.1), but with a sampling decoder. We compare *PDDN* with sampling but with no diversity component ( $D = 1$ ) to *PDDN* with sampling with higher values of  $D$  and observe that the diversity parameter is able to introduce additional diversity beyond the sampling decoder component.

To analyze the behavior of the diversity parameter  $D$  on the accuracy/validity and novelty trade-off in the drug domain, we generated samples for the FDA approved test set (Section 4.2), with various configurations of the diversity parameter  $D$ . Figure 3 presents the results for the two types of decoder functions. As we hypothesized, with both decoders, increasing

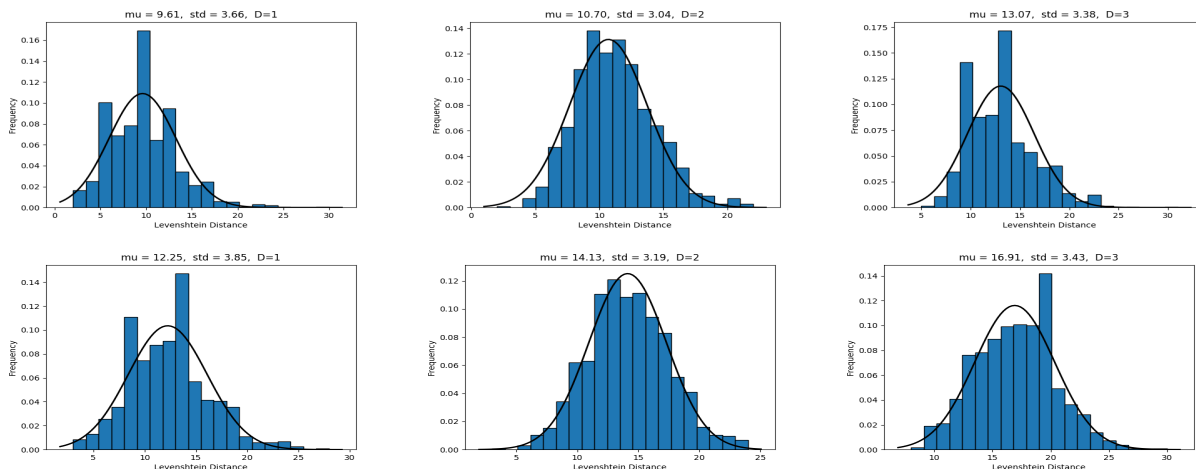


Figure 4: Levenshtein distance histograms for analyzing the diversity generated by *PDDN*. Top - Origin molecule vs Generated molecule distances. Bottom - within generated molecule population distances.

the value of the diversity parameter  $D$ , significantly increases the amount of novel molecules generated. As we expected, the novelty is not free, we observe lower accuracy and lower valid rates for increased diversity. Comparing argmax with sampling decoders, we observe that in general, sampling has lower accuracy and valid rate, but for low diversity value the sampling method generates significantly more novelty than the argmax. This behavior reduces for higher values of the diversity parameter, where both methods generate similar rates of novelty. We also observe the novelty rate reduces at some point of increased diversity value. This is quite expected because for large values of diversity, the latent molecule representation sampled with larger noise, thus at some point the generator is not able to recover much valid molecules in general, and novel ones in particular.

## 5.5 Molecular Variations

We would like to analyze not only whether a molecule is different from the prototype molecule but also quantify the diversity of the molecules with respect to the prototype molecule. Additionally, we would like to validate that the generated molecules originating from a prototype are also diverse with respect to one another. We compare the Levenshtein distances of the



Table 7: In class and cross drugs normalized distances computed on various drug classes.

Class	Cosine	L2	L1
Thiazide Diuretics	0.872	.95	.908
Benzodiazepines	0.923	.883	.859
$\beta$ -Blocker	0.866	.849	.822
NSAIDs	0.955	.853	.833
Cross Drugs	1.0	1.0	1.0

generated SMILES within the generated population and with respect to the prototype, used as input for a specific generation instance. We apply *PDDN* on the drug-like test set as prototypes. We note that we count only valid molecules generated in all evaluations. Figure 4 presents histograms of the Levenshtein distances for the generated molecules, with approximated Gaussian parameters and curve on top of the histograms. The top row represents the input prototype compared to the generated molecules Levenshtein distance distribution for different configurations of the diversity parameter  $D$  (increasing  $D$  from left to right). The bottom row represents the inner-generated population Levenshtein distance distribution for various values of  $D$ . On both type of distance evaluations (rows), we observe significantly larger Levenshtein distances for larger values of  $D$ , thus indicating positive effect of the diversity parameter  $D$  on both the distance from the prototype molecule, and the average inner distance between molecules that were generated with different random samples to the same prototype. Additionally, we observe *PDDN* diversity in generation is not limited to generating diversity with respect to the origin molecule, but also generates diversity within the generated population for a specific prototype, with higher amount of diversity tuned with the diversity parameter  $D$ .

## 5.6 Molecule Representation in Latent Space

Encoder-decoder settings produce intermediate representations of their input. In this section, we analyze the quality of those representations.

During *PDDN* generation process, we first encode molecule into a low dimensional vector

space with the encoder function. We refer to the output vector as the molecule embedding. To evaluate the embeddings, we leverage them for the task of drug classification. Intuitively, if the embeddings captures enough information for drug classification, we might rely on this representation for molecule generation. We note that for the task of encoding the molecule feature representation, we set the diversity parameter  $D = 1$ , but one should remember that the representation is still instantiated from unit Gaussian, and thus is not deterministic.

A drug class is a set of medications that have similar chemical structures, or the same mechanism of action (i.e., bind to the same biological target). In Table 7, we report embedding vector distances in the drug classes domain. We measure the average distance between all drug pairs belonging to the same drug class (i.e, “in class distance”) and normalize by the average distance between all drug pairs (i.e. “cross classes distance”). Thiazide and Benzodiazepines are chemical classes while  $\beta - Blocker$  and NSAIDs<sup>3</sup> are classes representing mechanism of action. We observe all in-classes distances are significantly lower than cross class. We conclude that although our molecule representation is noisy by the stochastic nature of *PDDN*, similarities in the embedding space are able to reflect significant similarities among various drug class.

## 6 Conclusions

Drug discovery is the process of identifying potential molecules that can be targeted for drugs. Common methods include systematic generation and testing of molecules via HTS. However, the molecular space is very large. Additional approaches require chemist to identify potential drugs based on their knowledge. Usually, one would start from a known compound in nature or known drug and identify potential changes. Approaches in machine learning today mainly focused on non-controlled molecule generation using generative mechanisms, such as VAE. The approaches were limited in their ability to generate both valid and novel molecules. In this work, we presented a prototype-based approach for generating drug-like

---

<sup>3</sup>NSAIDs - Nonsteroidal anti-inflammatory agents

molecules. We adopt the chemist approach of “borrowing” from nature or focusing on known drugs. We hypothesize that biasing the molecule generation towards known drugs might yield valid molecules. We train our model on drug-like molecules, and during generation extend VAE to, intuitively, search closer to the prototype (which can be a drug). We add additional component to diversify the molecules generated. We present results that show that many of the molecules generated are both valid and novel. When conditioning on drugs, we observe our system was able to generate known drugs that it never encountered before.

## References

- (1) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of computer-aided molecular design* **2013**, *27*, 675–679.
- (2) Garattini, S. Are me-too drugs justified? *Journal of Nephrology* **1997**, *10*, 283–294.
- (3) Schneider, G. Automating drug discovery. *Nature Reviews Drug Discovery* **2017**,
- (4) Schnecke, V.; Boström, J. Computational chemistry-driven decision making in lead generation. *Drug discovery today* **2006**, *11*, 43–50.
- (5) Lipinski, C. A. Drug-like properties and the causes of poor solubility and poor permeability. *Journal of pharmacological and toxicological methods* **2000**, *44*, 235–249.
- (6) Kingma, D. P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**,
- (7) Larsen, A. B. L.; Sønderby, S. K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300* **2015**,
- (8) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.;

- Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science* **2016**,
- (9) Organization, W. H. The selection and use of essential medicines: report of the WHO Expert Committee, 2002:(including the 12th model list of essential medicines). **2003**,
- (10) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*. 2012; pp 1097–1105.
- (11) Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015; pp 1–9.
- (12) Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*. 2013; pp 3111–3119.
- (13) Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* **2012**, *29*, 82–97.
- (14) Segler, M. H.; Preuss, M.; Waller, M. P. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature* **2018**, *555*, 604.
- (15) Mamoshina, P.; Vieira, A.; Putin, E.; Zhavoronkov, A. Applications of deep learning in biomedicine. *Molecular pharmaceutics* **2016**, *13*, 1445–1454.
- (16) Korotcov, A.; Tkachenko, V.; Russo, D. P.; Ekins, S. Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets. *Molecular pharmaceutics* **2017**, *14*, 4462–4475.

- (17) Aliper, A.; Plis, S.; Artemov, A.; Ulloa, A.; Mamoshina, P.; Zhavoronkov, A. Deep learning applications for predicting pharmacological properties of drugs and drug repurposing using transcriptomic data. *Molecular pharmaceutics* **2016**, *13*, 2524–2530.
- (18) Gao, X.; Qian, Y. Prediction of multi-drug resistant TB from CT pulmonary Images based on deep learning techniques. *Molecular pharmaceutics* **2017**,
- (19) Ramsundar, B.; Kearnes, S.; Riley, P.; Webster, D.; Konerding, D.; Pande, V. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072* **2015**,
- (20) Coley, C. W.; Barzilay, R.; Green, W. H.; Jaakkola, T. S.; Jensen, K. F. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of chemical information and modeling* **2017**, *57*, 1757–1772.
- (21) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: toxicity prediction using deep learning. *Frontiers in Environmental Science* **2016**, *3*, 80.
- (22) Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. O. ChemNet: A Transferable and Generalizable Deep Neural Network for Small-Molecule Property Prediction. *arXiv preprint arXiv:1712.02734* **2017**,
- (23) Kuzminykh, D.; Polykovskiy, D.; Kadurin, A.; Zhebrak, A.; Baskov, I.; Nikolenko, S.; Shayakhmetov, R.; Zhavoronkov, A. 3D Molecular Representations Based on the Wave Transform for Convolutional Neural Networks. *Molecular pharmaceutics* **2018**,
- (24) Wallach, I.; Dzamba, M.; Heifets, A. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855* **2015**,
- (25) Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*. 2015; pp 2224–2232.

- (26) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science* **2018**, *9*, 513–530.
- (27) Schwaller, P.; Gaudin, T.; Lanyi, D.; Bekas, C.; Laino, T. " Found in Translation": Predicting Outcome of Complex Organic Chemistry Reactions using Neural Sequence-to-Sequence Models. *arXiv preprint arXiv:1711.04810* **2017**,
- (28) Bjerrum, E. J. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076* **2017**,
- (29) Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. *Advances in Neural Information Processing Systems*. 2017; pp 2604–2613.
- (30) Kadurin, A.; Nikolenko, S.; Khrabrov, K.; Aliper, A.; Zhavoronkov, A. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular pharmaceutics* **2017**, *14*, 3098–3104.
- (31) Kadurin, A.; Aliper, A.; Kazennov, A.; Mamoshina, P.; Vanhaelen, Q.; Khrabrov, K.; Zhavoronkov, A. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* **2017**, *8*, 10883.
- (32) Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *arXiv preprint arXiv:1701.01329* **2017**,
- (33) Ikebata, H.; Hongo, K.; Isomura, T.; Maezono, R.; Yoshida, R. Bayesian molecular design with a chemical language model. *Journal of computer-aided molecular design* **2017**, *31*, 379–391.

- (34) Neil, D.; Segler, M.; Guasch, L.; Ahmed, M.; Plumbley, D.; Sellwood, M.; Brown, N. Exploring Deep Recurrent Models with Reinforcement Learning for Molecule Design. **2018**,
- (35) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *Journal of cheminformatics* **2017**, *9*, 48.
- (36) Cadeddu, A.; Wylie, E. K.; Jurczak, J.; Wampler-Doty, M.; Grzybowski, B. A. Organic chemistry as a language and the implications of chemical linguistics for structural and retrosynthetic analyses. *Angewandte Chemie* **2014**, *126*, 8246–8250.
- (37) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **1988**, *28*, 31–36.
- (38) Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* **2014**,
- (39) Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for De Novo Drug Design. *Molecular informatics* **2017**,
- (40) Ertl, P.; Lewis, R.; Martin, E.; Polyakov, V. In silico generation of novel, drug-like chemical matter using the LSTM neural network. *arXiv preprint arXiv:1712.07449* **2017**,
- (41) Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural computation* **1997**, *9*, 1735–1780.
- (42) Williams, R. J.; Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* **1989**, *1*, 270–280.
- (43) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling* **2010**, *50*, 742–754.

- (44) Koutsoukas, A.; Simms, B.; Kirchmair, J.; Bond, P. J.; Whitmore, A. V.; Zimmer, S.; Young, M. P.; Jenkins, J. L.; Glick, M.; Glen, R. C. From in silico target prediction to multi-target drug design: current databases, methods and applications. *Journal of proteomics* **2011**, *74*, 2554–2574.
- (45) Jenkins, J. L.; Bender, A.; Davies, J. W. In silico target fishing: Predicting biological targets from chemical structure. *Drug Discovery Today: Technologies* **2006**, *3*, 413–421.
- (46) Alvarsson, J.; Eklund, M.; Engkvist, O.; Spjuth, O.; Carlsson, L.; Wikberg, J. E.; Noeske, T. Ligand-based target prediction with signature fingerprints. *Journal of chemical information and modeling* **2014**, *54*, 2647–2653.
- (47) Bento, A. P.; Gaulton, A.; Hersey, A.; Bellis, L. J.; Chambers, J.; Davies, M.; Krüger, F. A.; Light, Y.; Mak, L.; McGlinchey, S. The ChEMBL bioactivity database: an update. *Nucleic acids research* **2014**, *42*, D1083–D1090.
- (48) Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. TensorFlow: A System for Large-Scale Machine Learning. OSDI. 2016; pp 265–283.
- (49) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,
- (50) Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010; pp 249–256.
- (51) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling* **2012**, *52*, 1757–1768.



- (52) Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Woolsey, J. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research* **2006**, *34*, D668–D672.
- (53) Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*. 2014; pp 3104–3112.
- (54) Landrum, G. RDKit: Open-source cheminformatics. *Online*). <http://www.rdkit.org>. Accessed **2006**, *3*, 2012.