molecular pharmaceutics

Article

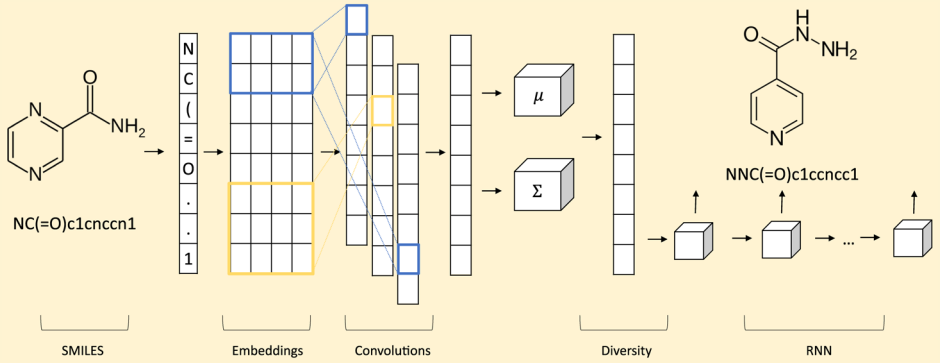
Cite This: *Mol. Pharmaceutics* 2018, 15, 4406−4416

pubs.acs.org/molecularpharmaceutics

Prototype-Based Compound Discovery Using Deep Generative Models

Shahar Harel* and Kira Radinsky*

Department of Computer Science, Technion - Israel Institute of Technology, Haifa 3200003, Israel



ABSTRACT: Designing a new drug is a lengthy and expensive process. As the space of potential molecules is very large (Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 675−679 10.1007/s10822-013-9672-4), 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*

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.[2]

During this process of prototype-based drug discovery, an interdisciplinary team of scientists generates a 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.[3] They rely on their expertise and medicinal chemistry intuition to create chemical hypotheses, which have been shown to be biased.[4]

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

Special Issue: Deep Learning for Drug Discovery and Biomarker Development

Received: May 6, 2018
Revised: July 19, 2018
Accepted: July 31, 2018
Published: July 31, 2018

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

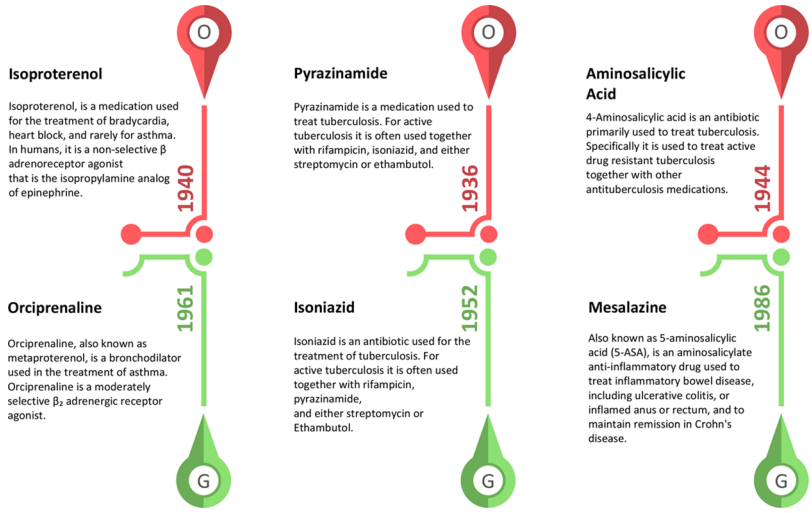


Figure 1. Drug development timeline, with an example of drugs generated by PDDN (bottom), using FDA approved drugs as prototypes (top).

It is common to encode molecular structures into SMILES notations (simplified molecular-input line-entry system), which preserve 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, which is a common chemical drug-design qualitative measure that estimates the structure bioavailability, solubility in water, potency, etc.[5]

Variational auto encoders (VAE)[6] are an 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.[6,7] Training a vanilla VAE on drug-like molecules provides an ability to sample new molecules which, intuitively, should be drug-like.[8] In this work, we extend VAE to allow a conditional sampling, which is sampling an example from the data distribution (drug-like molecules) which is closer to a given input. This allows for sampling molecules closer to a prototype drug, and thus increases the probability of generating a valid drug with similar characteristics. Additionally, we add a diversity component that allows for 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 for 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 a high percentage of valid molecules.

Additionally, we perform retrospective experiments and use drugs developed in the 1930s and 1940s as prototypes. The system was then able to generate new drugs, some of which were 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".[9] In the retrospective experiment, we used as prototypes only drugs discovered until 1940. For the drug pyrazinamide, 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.

■ RELATED WORK

Over the past decade, deep neural networks (DNN) have been a game changer mainly in a few areas of machine learning research, such as computer vision,[10,11] natural language processing,[12] and speech recognition.[13] More recently, deep learning is rapidly advancing in many more areas of science, such as computational chemistry and pharmacology.[14−18] DNN most prominent success stories are observed in domains with access to large, raw (unprocessed) data sets. 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.[19−21] Such approaches are semioptimal as they restrict the space of potential representations through the assumptions made by limiting to the chosen features.[22]

More recent methods overcome this issue by leveraging advanced DNN models to learn chemical continuous representations (i.e., embeddings) based on large data sets 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 that served as input to a convolutional neural network (CNN),[23,24] molecular graph representation paired with neural graph embedding methods,[25,26] and SMILES strings, which were modeled as a language model with a recurrent neural network (RNN).[8,27,28] Others[20,27,29] leverage the embeddings for numerous supervised prediction tasks, 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

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

discovery.[8,30,31] Most methods aim at generating valid molecules. For example, Segler et al. trained 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[8] leverage the VAE generative model[6] 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[32] extend this approach by tuning the model on a data set of molecules with specific characteristics or by applying post-processing methods, such as bayesian optimization[8,33] and reinforcement learning.[34,35]

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 (Experiments section), 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.

METHODS

We define the problem of a 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 like 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 a prototype-based generation with a diversification factor. We start by reviewing how molecules are represented as text (Molecule Representation section) and then present a generative model (Molecule Driven Hypothesis Generation section). Our generative model builds upon recent methods for deep representation learning. We train a stochastic neural network to learn internal molecule representation (embedding). After the molecule embedding was obtained, we further utilized the stochastic component of the neural architectures to introduce a parametrized diversity layer into the generation process. The architecture of our proposed solution is presented in the PDDN Architecture section.

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.[36] 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 linguistic-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.[37] 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.[8,27,28,32] 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.

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.[6,7] One of the most popular generative frameworks is VAE.[6]

VAE are encoder−decoder models that use a variational approach for latent representation learning. Intuitively, the encoder was constrained to generate latent representations that follow a prior. During generation, latent vectors were sampled from the priors and passed to the decoder that generates the new representation. We leveraged VAE for the task of molecule generation. The stochasticity allowed integrating chemical diversity into the generation process. However, application of generative models for molecule generation have shown limited results.[8] Unlike image generation, where each image is valid, when we aimed at molecule generation, each representation was not a valid molecule representation. Intuitively, when we sampled from the prior for image generation, the space of the images was much more dense than that of valid molecules. Therefore, many image samples were valid compared to randomly generated molecule representations. We hypothesized that a constrained generation next to a known prototype, rather than a nonconstrained sampling, would yield a better molecule generation. We extended the VAE generation process to conditions on a prototype, i.e., generate molecules closer to a given drug. Intuitively, directing the sampling process closer to existing prototype drugs might have yielded valid molecules that carry similar characteristics to the prototype yet provide diversity. Our results provided evidence that a conditioned sample alongside a diversity component yielded more valid and novel results. If conditioned on known drugs, the system was able to generate drugs discovered years after the prototype (Introduction section).

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

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

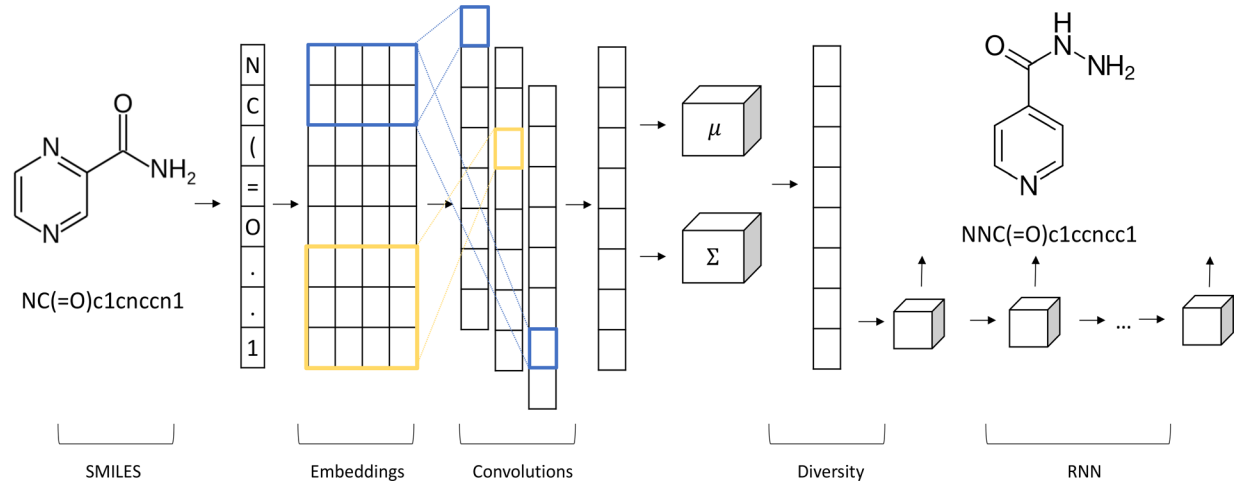


Figure 2. PDDN end-to-end neural net architecture.

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

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

where q is approximated via an encoder neural network function, applied on molecule M as input, and outputs the latent feature parametrization $(\hat{\mu},\hat{\sigma})$ of the molecule. We then sampled instances from this parametrization to obtain the final encoded output z; p is represented via a 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 reflected multinomial distribution over the atoms space. Each atom was represented via a character. We formed the character generation process as an iterative process, each character y_i was generated on the basis of the hidden encoded representation z and the formerly generated y_i's. In total, the output of this step was a string $\hat{y}_i = \{\hat{y}_1,..,\hat{y}_N\}$, where N is a predefined maximal generation length. Formally, for a single character y_i

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

where y_{i-1} is the character embedding corresponding to the last character generated, and s_i is a state at step i representing the current processed information on 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 the generation process, we introduced 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 instantiated our encoder output parameters with a diversity layer. Intuitively, the diversity layer output a noisy sample from a distribution centered as the encoder suggested but with a larger variance. This allowed us to explore the molecule space around an origin molecule, with a tune-able amount of diversity, corresponding to the variability in chemical space. The diversity layer sampled a noisy instance according to the encoded Gaussian parameters and a diversity parameter D.

The output of the diversity layer was 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 noise sample n, from Gaussian distribution with diversity parameter D, $[n \sim \mathcal{N}(O, D)]$

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

We obtained an 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 was set to 1. Thus, the z instance was sampled from the nondiverse distribution suggested by the computed parameters. Tuning this parameter at a generation time allowed us to explore the space around the prototype.

PDDN Architecture. We leveraged 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 the PDDN architecture.

PDDN started by encoding the molecule (in SMILES notation) using the encoder function. First, it encoded each character in the SMILES representation into its d dimensional embedding, and then it applyied convolutions over various substring (filter) sizes (e.g., corresponding to chemical substructures). A similar encoder architecture was suggested for NLP tasks, such as sentence classification.[38] The extracted features were then concatenated, and fully connected layers were applied. The outputs of the encoder were considered as a vector of means and a vector of standard deviations, representing the distributions of features for the prototype. In VAE, those vectors were then fed into a decoder. The goal was to optimize reconstruction of the original input and constrain the representation to a known prior. During generation, the vectors of features were sampled from the prior distribution, and their output was passed to the decoder that generated a new representation.

We extended the VAE generation process by adding a diversity layer. During generation, instead of sampling from the prior means and standard deviations, we first fed a prototype. We sampled from the prototype feature distribution with parametrized diversity (Molecule-Driven Hypothesis Generation section) to form the prototype latent representation, which served as input for the decoder.

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

As described in the Molecule-Driven Hypothesis Generation section, our decoder was a sequential generator. By generating sequentially, we formed another parameter of variability in the generated data by introducing minor variations into the molecule generated during generation. This was the main component of many other works on molecule generation[32,39,40] to introduce diversity into the generation process. We later showed that our diversity layer could introduce diversity beyond this component.

We represented our decoder as a RNN (LSTM[41]). The decoder received the encoder output as its input. The encoded representation formed the first state of the decoder. The decoder then generated 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 the former step.

During training, we fed the decoder with the correct next symbol, even if it was wrongly predicted.[42] During generation, we experimented with two options for generating the next symbol: one by selecting the best scored character from the distribution over symbols (argmax), and the second was by sampling from the same distribution. By introducing sampling into the generation, we were able to increase the amount of variability we generated during generation. The model was 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 used the cross-entropy loss, defined as

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

We note that, we minimized the variational lower bound,[6] which was 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 included keeping track of atoms, rings, and brackets to eventually close them. In this case, the lower dimension representation that could be reconstructed into a valid prototype is a highly informative representation. In other words, by minimizing the reconstruction error, we learned a prototype continuous representation that captured the coordinates along the main factors of variation in the chemical space. This representation was the base for further diversifying the molecule generation process.

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 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 (log*P*, molar refractivity etc.). In this work, we developed two methods for ranking the generated population with respect to a prototype input.

(1) Lipinski-based ranking: We ranked the generated molecules in proportion to the number of violations to Lipinski's rule of five.[5] This rule is a heuristic that evaluates the drug-likeness 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 possess. During our ranking, we therefore preferred candidates that carried a resemblance to the prototype drug. We represented each molecule via a fingerprint vector representation[43] and used it as a secondary ranking criteria, the Tanimoto coefficient, to calculate molecular similarity between the generated compounds and the prototype (i.e., a generated molecule with the least violations to the Lipinski rule and the most similar to the prototype was ranked higher).

(2) Target-based ranking: It is common[44,45] 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.[43] Following this observation and our goal of generating molecules with similar characteristics to a prototype, we built a ranking mechanism that emphasized the prototype target binding profile. We used a priorly trained ligand-based target prediction model[46] obtained from ChEMBL[47] to obtain the probability vector over approximately 1500 targets for both the prototype and each of the generated compounds. Finally, we ranked the generated molecules in proportion to the cosine similarity with the input prototype, which is represented by the target probabilities vector.

EXPERIMENTAL SETTINGS

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

Model Details. PDDN was trained using a Tensorflow API.[48] We used the Adam algorithm[49] to optimize all of 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 a truncated normal with std 0.1, and all other weights used the Xavier initialization,[50] and biases were initialized with the constant. To reduce overfitting, we included an early stopping criteria based on the validation set reconstruction error. We used the exponential decay factor on the learning rate, and the teacher forcing method[42] during training. In total, Table 1 presents PDDN hyperparameter configuration.

The code for our system is available over github (https://github.com/shaharharel/CDN_Molecule) for further research in the community

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

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

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	[a]	0.58				
PDDN $D = 1$	0.91	0.89	0.19	0.9	0.89	8
PDDN $D = 2$	0.82	0.81	**0.26**	0.81	0.8	**66.6**
PDDN $D = 3$	0.64	0.63	**0.37**	0.65	0.65	**227**

[a]As no prototype is given, there is no reconstruction to measure.

Data Sets. *Drug-Like Molecules Database.* In our work, we provide experiments showing that PDDN is capable of generating drug-like molecules. We train our model on a large drug-like molecule database and present several metrics on the generated molecules. The ZINC database[51] contains commercially available compounds for structure-based virtual screening. In addition, the databases have subsets of ZINC filtered by physical properties. One such filtering is based on Lipinski's rule of five,[5] which is 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 200 000 drug-like compounds extracted at random from the ZINC drug-like database. The subset was further divided into train/validation/test sets, with 5000 compounds for the validation and test sets, and the rest for the training set. The subsets were used for training the model (train), evaluating hyperparameters and stopping criteria (validation), and for method evaluation and experiments (test).

Drug Database. For our drug generation experiment (Drug Generation section), we showed that some of the molecules generated by PDDN were drugs which were discovered years later. The DrugBank database[52] 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.

Compared Methods. As discussed in the Related Work section, 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 postprocessing to achieve general molecular characteristics. For example, Olivecrona et al.[35] applied reinforcement learning over a generated molecule to resemble another predefined molecule. We compared 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:[53] An auto-encoder architecture was applied on sequence data for prediction of sequences. Both encoder and decoder were RNN. Although the model is in general deterministic, it was 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 considered two baselines, one using the Argmax method and the other utilizing the sampling method to reach diversity.
(2) Conv2Seq: To conform better with the PDDN parameter setting that is utilized on CNNs, we implemented a second auto-encoder the same as the previous method but with a convolution encoder.
(3) VAE:[6] A vanilla implementation of VAE. This model generated new molecules from unit Gaussian random samples, regardless of prototypes.
(4) PDDN−VAE: Our diversity model on top of a variational auto-encoder. D is the diversity parameter of eq 4 The higher the D, the higher the diversity induced. We note that for $D = 1$, the model extended VAE for a conditional setting but without diversity.

EXPERIMENTS

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

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 examined the methods on the task of prototype reconstruction on a test set of 5000 ZINC drug-like compounds. To explicitly address the reconstruction accuracy and validity vs the generated molecules diversity, we measured the following metrics:

(1) Reconstruction accuracy (acc): Character-level accuracy with the input prototype serving also as a target.
(2) Valid molecule percentage (valid): Percentage of valid molecules. There are several numeric validations performed on a molecule representation to validate its correctness. We used the Rdkit[54] library to measure the 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 measured all of the above metrics with @k notation. In our context, @k represents that for each prototype compound, we ran the PDDN generation process with k instances of random noises

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

Table 3. Sample of Automatically Generated Drugs and the Drug Served as a Prototype to the Generation Process

input drug	input SMILES	generated drug	generated SMILES
aminosalicylic	Nc1ccc(C(=O)O)c(O)c1	mesalazine	Nc1ccc(O)c(C(=O)O)c1
pyrazinamide	NC(=O)c1cnccn1	isoniazid	NNC(=O)c1ccncc1
protriptyline	CNCCCC1c2ccccc2C=Cc2ccccc21	desipramine	CNCCCN1c2ccccc2CCc2ccccc21
phenelzine	NNCCc1ccccc1	isoniazid	NNC(=O)c1ccncc1
isoproterenol	CC(C)NCC(O)c1ccc(O)c(O)c1	orciprenaline	CC(C)NCC(O)c1cc(O)cc(O)c1
pheniramine	CN(C)CCC(c1ccccc1)c1ccccn1	tripelennamine	CN(C)CCN(Cc1ccccc1)c1ccccn1

parametrized with diversity D. We note that, to measure novel@k, we had to count how many unique molecules were generated; a novel molecule was counted only once, even if it was generated with various Gaussian samples for the prototype. The metric of novel@k was 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 observed that with a diversity level of 1 (nondiverse sampling), PDDN generated a similar diversity to the baselines. Increasing D, significantly increased 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 observed that PDDN was 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 molecules significantly increasing with the diversity parameter D.

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

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

Evaluation on this task was harder since our goal was to generate drugs, and we could not a-priori know if the generated molecule had the desired characteristics of a drug without further experimenting with the compound. We therefore considered as gold standard a test set of 869 approved known drugs. Although this test set is very small in comparison with the enormous molecule space, some approved drugs are chemically similar and share similar therapeutic characteristics; thus, we hypothesized 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 targeted existing drugs as prototypes by feeding them one-by-one in a leave-one-out cross validation manner to PDDN. The generated molecules were 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.

Retrospective Generation Experiments. We ran the baselines and PDDN variants over all 869 approved drugs data set 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 of 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 observed that the VAE could not produce any known drug. We hypothesized that this stems from the fact that VAE randomly generates a molecule and is not based on a prototype. PDDN with no diversity and the other baselines generated 9−12 unique drugs. This result emphasized how the variability that the decoders present during sampling contributes to the generation of known drugs. More interestingly, we observed that for higher D values of our diversity layer (PDDN−VAE D = 2 and PDDN−VAE D = 3), the amount of unique known drugs increased significantly.

Table 4. Automatically Generated FDA-Approved Drugs[a]

model	# drugs	% from generated valid molecules
VAE	0	0
Seq2Seq	12	0.002
Conv2Seq	9	0.0018
PDDN-VAE D = 1	12	0.0023
PDDN-VAE D = 2	22	0.005
PDDN-VAE D = 3	35	**0.01**

[a]We present the percentage of the FDA-approved drug from the total valid molecules generated by each method.

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 resulted in samples that do not coincide with the prior distribution enforced during training over the latent space (Molecule-Driven Hypothesis Generation section), unlike VAE and PDDN with D = 1. Thus, such sampling resulted in final marginal distribution that did not match the prior, which in turn resulted in a lower number of valid molecules. However, we were more interested in novel molecules and medical discoveries, even on the expanse of valid molecules. Thus, with the idea of exploring the latent space, we added 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

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

Table 5. Percision@10/100 of Automatically Generated FDA-Approved Drugs Using PDDN for Generating and Ranking the Molecules[a]

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
PDDN−VAE $D = 1$	11	12	10	12
PDDN−VAE $D = 2$	19	22	19	22
PDDN−VAE $D = 3$	23	31	17	31

[a]L stands for Lipinsiki 5-based ranking and TP for target prediction-based ranking.

$D = 1$ 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.

Retrospective Candidate Ranking Experiments. Table 5 presents the generation and ranking performance of PDDN with both ranking methods (see Ranking the Generated Population section). 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 observed that both ranking methods were able to capture most of the generated drugs within its top ranked candidates. We observed 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 observed an additional phenomena, although the higher the diversity, the higher the amount of novel-generated molecules (Table 2), the precision@10 and precision@100 of those methods are higher. This indicated 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 does not 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 a prototype to the generation process, our model was able to chemically diversify the prototype drug in a way that generated another known drug. We were encouraged by the results that the PDDN was able to generate a significant number of previously known drugs during the retrospective experiment.

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 eliminated all known drugs from the ZINC database and those that were presented as prototypes of 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 drugs 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). An 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, which was used to treat inflammatory bowel disease based on an antibiotic primarily used to treat tuberculosis that was discovered about 40 years before.

Diversity Mechanisms. A common method to employ diversity in encoder−decoder models is to employ a sampling decoder into the architecture.[32,39,40] The diversity was introduced by sampling from distribution over characters in each time step of generation, rather than choosing the topmost (argmax) character at test time. We analyzed the contribution of the diversity layer D for PDDN presented in this work alongside a sampling decoder as well. Table 6 presents the PDDN performance on the previous metrics (Novel Molecules Generation section) but with a sampling decoder. We compared PDDN with sampling, but with no diversity component ($D = 1$) to PDDN with sampling with higher values of D, and observed that the diversity parameter was able to introduce additional diversity beyond the sampling decoder component.

Table 6. PDDN Performance Using a Sampling Decoder

model	acc	valid	novel	A@1k	V@1k	N@1k
PDDN $D = 1$	0.88	0.78	0.19	0.88	0.79	39.5
PDDN $D = 2$	0.8	0.69	**0.25**	0.79	0.68	**94**
PDDN $D = 3$	0.57	0.39	**0.27**	0.56	0.38	**179**

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

Molecular Variations. We wanted to analyze not only whether a molecule is different from the prototype molecule

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

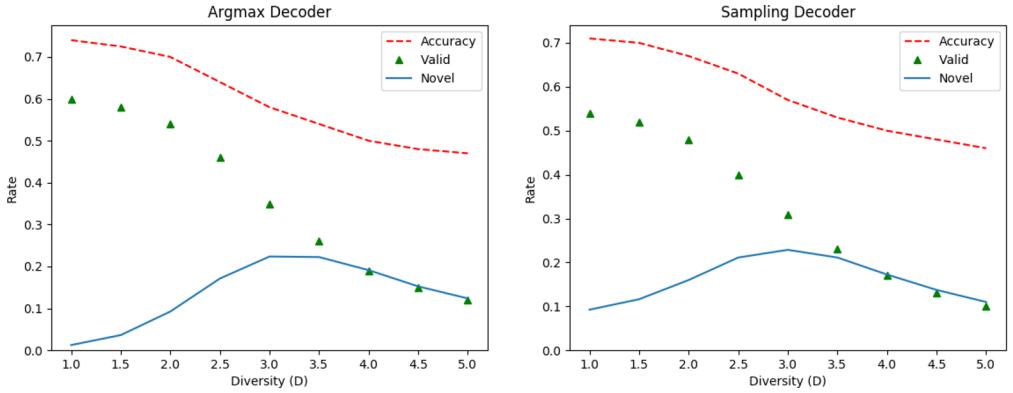


Figure 3. Diversity parameter effect on performance.

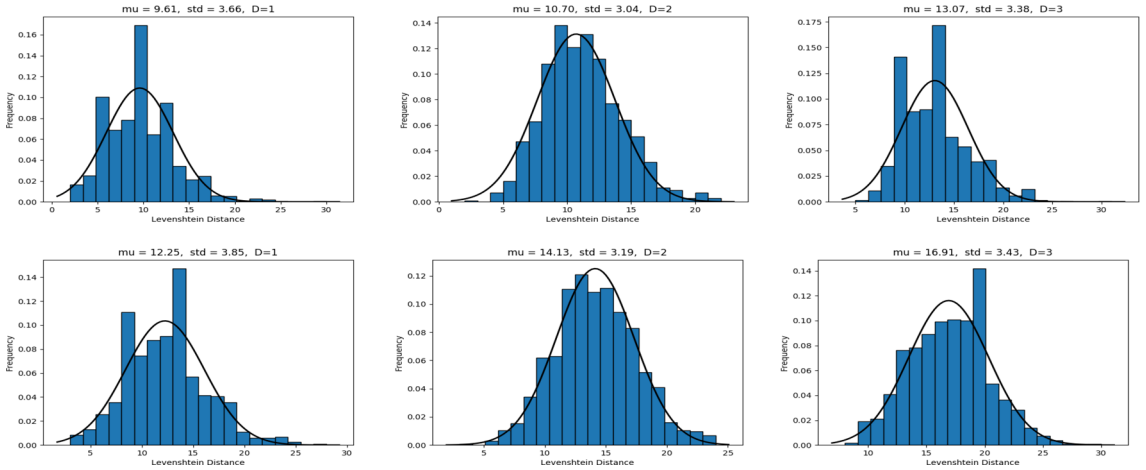


Figure 4. Levenshtein distance histograms for analyzing the diversity generated by PDDN. (Top) Origin molecule vs generated molecule distances. (Bottom) Within generated molecule population distances.

but also quantify the diversity of the molecules with respect to the prototype molecule. Additionally, we wanted to validate that the generated molecules originating from a prototype are also diverse with respect to one another. We compared the Levenshtein distances of the generated SMILES within the generated population and with respect to the prototype, which was used as input for a specific generation instance. We applied PDDN on the drug-like test set as prototypes. We note that we counted only valid molecules generated in all evaluations. Figure 4 presents histograms of the Levenshtein distances for the generated molecules, with approximated Gaussian parameters and a 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 types of distance evaluations (rows), we observed significantly larger Levenshtein distances for larger values of D, indicating a positive effect of the diversity parameter D on both the distance from the prototype molecule and the average inner distances between molecules that were generated with different random samples to the same prototype. Additionally, we observed that PDDN diversity in generation was not limited to generating diversity with respect to the origin molecule, but also it generated diversity within the generated population for a specific prototype, with a higher amount of diversity tuned with the diversity parameter D.

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 the PDDN generation process, we first encoded a molecule into a low dimensional vector space with the encoder function. We refered to the output vector as the molecule embedding. To evaluate the embeddings, we leveraged them for the task of drug classification. Intuitively, if the embeddings captured enough information for drug classification, we might have had to 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 was still instantiated from unit Gaussian, and thus was 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 measured the average distance between all drug pairs belonging to the same drug class (i.e, "in class distance") and normalized them by the average distance between all drug pairs (i.e., "cross classes distance"). Thiazide and benzodiazepines are chemical classes, while β-blocker and NSAIDs (nonsteroidal anti-inflammatory agents) are classes representing the mechanism of action. We observed that all in-classes distances were significantly lower than those of cross-class. We

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406–4416

Table 7. In Class and Cross Drugs Normalized Distances Computed on Various Drug Classes

class	cosine	L2	L1
thiazide diuretics	0.872	0.95	0.908
benzodiazepines	0.923	0.883	0.859
β-blocker	0.866	0.849	0.822
NSAIDs	0.955	0.853	0.833
cross drugs	1.0	1.0	1.0

concluded that although our molecule representation was noisy by the stochastic nature of PDDN, similarities in the embedding space were able to reflect significant similarities among various drug classes.

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 a 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 are mainly focused on noncontrolled molecule generation using generative mechanisms, such as VAE. The approaches are limited in their ability to generate both valid and novel molecules. In this work, we presented a prototype-based approach for generating drug-like molecules. We adopted the chemist approach of "borrowing" from nature or focusing on known drugs. We hypothesized that biasing the molecule generation toward known drugs might yield valid molecules. We trained our model on drug-like molecules, and during generation, extended VAE to, intuitively, search closer to the prototype (which can be a drug). We added an 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 drugs, we observed that our system was able to generate known drugs that it had never encountered before.

AUTHOR INFORMATION

Corresponding Authors

*E-mail: sshahar@cs.technion.ac.il.

*E-mail: kirar@cs.technion.ac.il.

ORCID

Shahar Harel: 0000-0001-8175-5031

Notes

The authors declare no competing financial interest.

REFERENCES

(1) Polishchuk, P. G.; Madzhidov, T. I.; Varnek, A. Estimation of the size of drug-like chemical space based on GDB-17 data. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 675−679.

(2) Garattini, S. Are me-too drugs justified? *J Nephrol.* **1997**, *10*, 283−294.

(3) Schneider, G. Automating drug discovery. *Nat. Rev. Drug Discovery* **2017**,.*1797*

(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. *J. Pharmacol. Toxicol. Methods* **2000**, *44*, 235−249.

(6) Kingma, D. P.; Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* **2013**, *2013*, 1.

(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**, *2015*, 1.

(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 Cent. Sci.* **2018**, *4* (2), 268.

(9) WHO. *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**, 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**, 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.; Kingsbury, B. 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. *Mol. 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. *Mol. 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. *Mol. 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. *Mol. Pharmaceutics* **2018**, 1.

(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**, *2015*, 1.

(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. *J. Chem. Inf. Model.* **2017**, *57*, 1757−1772.

(21) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: toxicity prediction using deep learning. *Front. Environ. Sci.* **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**, *2017*, 1.

(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. *Mol. Pharmaceutics* **2018**, 1.

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416

(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**, *2015*, 1.
(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**, 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. *Chem. Sci.* **2018**, *9*, 6091.
(28) Bjerrum, E. J. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076* **2017**, *2017*, 1.
(29) Jin, W.; Coley, C.; Barzilay, R.; Jaakkola, T. Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network. *Advances in Neural Information Processing Systems* **2017**, 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. *Mol. 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. *ACS Cent. Sci.* **2018**, *4*, 120.
(33) Ikebata, H.; Hongo, K.; Isomura, T.; Maezono, R.; Yoshida, R. Bayesian molecular design with a chemical language model. *J. Comput.-Aided Mol. Des.* **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*, ICLR 2018 Conference; 2018.
(35) Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminf.* **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. *Angew. Chem.* **2014**, *126*, 8246−8250.
(37) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.
(38) Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* **2014**, *2014*, 1.
(39) Gupta, A.; Müller, A. T.; Huisman, B. J.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for De Novo Drug Design. *Mol. Inf.* **2018**, *37*, 1700111.
(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**, *2017*, 1.
(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. *J. Chem. Inf. Model.* **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.; Bender, A. From in silico target prediction to multi-target drug design: current databases, methods and applications. *J. 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: Technol.* **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. *J. Chem. Inf. Model.* **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.; et al. The ChEMBL bioactivity database: an update. *Nucleic Acids Res.* **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**, 265−283.
(49) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**, *2014*, 1.
(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. *J. Chem. Inf. Model.* **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 Res.* **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**, 3104−3112.
(54) Landrum, G. RDKit: Open-source cheminformatics. http://www.rdkit.org (accessed 2006.

DOI: 10.1021/acs.molpharmaceut.8b00474
Mol. Pharmaceutics 2018, 15, 4406−4416