Bidirectional Molecule Generation with Recurrent Neural Networks

Francesca Grisoni*, Michael Moret, Robin Lingwood, Gisbert Schneider*

ETH Zurich, Department of Chemistry and Applied Biosciences,
RETHINK, Vladimir-Prelog-Weg 4, 8093, Zurich, Switzerland

*francesca.grisoni@pharma.ethz.ch; gisbert.schneider@pharma.ethz.ch

Recurrent neural networks (RNNs) are able to generate de novo molecular designs using SMILES string representations of chemical structure. RNN-based structure generation is usually performed unidirectionally, by growing SMILES strings from left to right. However, there is no natural start or end of a small molecule, and SMILES strings are intrinsically non-univocal representations of molecular graphs. These properties motivate bidirectional structure generation. Here, bidirectional generative RNNs for SMILES-based molecule design are introduced. To this end, two literature-based bidirectional methods (Synchronous Forward Backward RNN, and Neural Autoregressive Distribution Estimator) were implemented, and a novel method for SMILES string generation is presented – the Bidirectional Molecule Design by Alternate Learning (BIMODAL). These three bidirectional strategies were compared to the unidirectional forward RNN approach for SMILES string generation, in terms of the (i) novelty, (ii) scaffold diversity, and (iii) chemical and biological relevance of the computer-generated molecules. The results of this study positively advocate bidirectional strategies for SMILES-based molecular *de novo* design, with BIMODAL showing superior results to the unidirectional forward RNN for most of the criteria. The program code of the methods and the pre-trained models can be found at URL https://github.com/ETHmodlab/BIMODAL.

This document is the unedited Author's version of a Submitted Work that was subsequently accepted for publication in the *Journal of Chemical Information and Modeling*, copyright © American Chemical Society after peer review. To access the final edited and published work see https://pubs.acs.org/doi/10.1021/acs.jcim.9b00943.

Introduction

The chemical space of synthesizable small organic molecules is estimated to contain 10^{60} to 10^{100} chemical structures.[1,2] Designing molecules with desired properties from scratch confronts chemists with a complex multivariate optimization task. Computational approaches have proved valuable to generate novel molecules,[3] *e.g.*, by extensive structure enumeration,[4,5] inversion of quantitative structure-activity relationship (QSAR) models,[6–8] evolutionary algorithms,[9,10] or rule-based design.[11,12] Most of these methods rely on *a priori* knowledge, *e.g.*, structure-activity relationships, aggregation rules, chemical transformation rules, fitness functions and/or design constraints. Recently, generative deep learning methods (*e.g.*, recurrent neural networks,[13,14] adversarial autoencoders[15]) have emerged as potential alternatives to rule-based *de novo* molecular design methods.[16–21] Many of these generative machine learning methods build on text representations of molecules, such as SMILES (Simplified Molecular Input Line Entry Systems,[22] Figure 1a) strings, and directly sample new chemical entities without the need of explicit design rules, structure-activity relationship models, or molecular descriptors.

Recurrent neural networks (RNNs), in particular, have been applied to computational molecule generation.[21,23–25] In a recent benchmark study,[26] SMILES-based RNN resulted the best generative method among a selection of evolutionary, rule-based, and sequence-based methods (*i.e.*, generative adversarial networks with reinforcement learning,[20] and adversarial autoencoders[19]), in the task of matching the training data distribution of structural and biological features. Moreover, RNNs proved applicable to generating new nuclear receptor modulators and bioactive natural-product inspired molecules in prospective studies.[27,28]

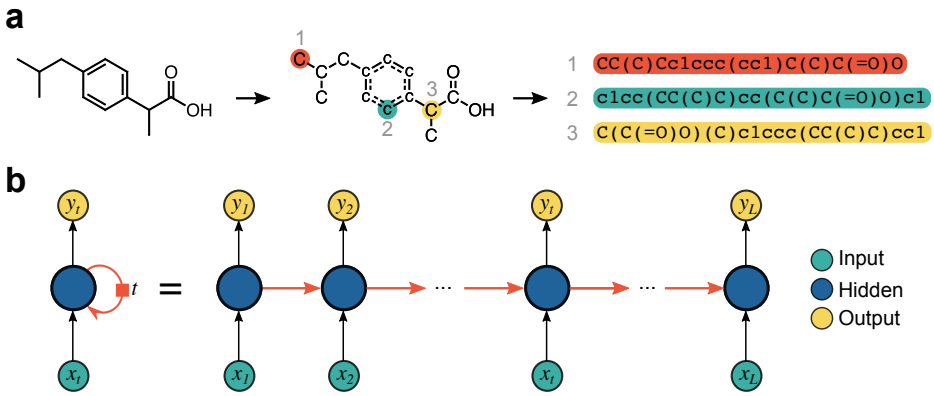

Figure 1. Overview of the basic concepts of this study. (a) Simplified Molecular Input Line Entry Systems (SMILES) strings, obtained from a molecular graph representation, where each atom is indicated by its element symbol, while branching and connectivity are indicated by symbols or lowercase letters (*e.g.,* '( )', '=' and 'c' for branching, double bonds and aromatic carbons, respectively). Examples of three SMILES strings representing the drug ibuprofen are shown; the start atoms used for SMILES string generation are indicated by grey numbers. (b) Simplified scheme of a forward recurrent neural network (RNN) with one recurrent neuron layer. RNNs model a dynamic system, in which the network state at any *t*-th time point depends both on the current observation (x_t) and on the previous state (at *t-1*), and is used to predict the output (y_t).

With sequences of SMILES characters ("tokens") as input, RNN models learn to predict one token at a time, based on the preceding portions of the sequence and a probability estimation (Figure 1b). From the learnt probability distribution, new SMILES strings can be sampled. RNNs are usually trained to read and generate SMILES strings in a "forward" manner, that is, from the left to the right. However, SMILES representations are non-univocal[22] (Figure 1a), as they can be written starting from any non-hydrogen atom and by proceeding in any direction. Also, there is no natural start and end of a small molecule. The non-univocity and non-directionality offers the opportunity to explore bidirectional sequence generation, that is, methods that read and generate SMILES strings in both forward and backward direction. To the best of our knowledge, bidirectional SMILES generation has not been explored to date.

The aim of the present study is to implement and evaluate bidirectional RNNs as SMILES string generators. To this end, two literature-based methods (synchronous forward backward RNN [FB-RNN],[29] neural autoregressive distribution estimator [NADE][30]), and the here newly developed method (bidirectional molecule design by alternate learning [BIMODAL]) were implemented and compared with a conventional forward RNN approach for SMILES string generation. The methods were evaluated with regard to their ability to construct chemically-relevant molecules, in terms of the novelty of their scaffolds and selected biological and physicochemical properties. The results of this study highlight the advantages of bidirectional sequence generation methods and advocate their usage for prospective *de novo* molecular design.

Theory

Forward recurrent neural networks

RNNs[31] can deal with sequence data as both input and output. Given an input vector at time step t, $\mathbf{x}(t) = \{x_1, x_2, \ldots x_t\}$, and the corresponding output vector $\mathbf{y}(t) = \{y_1, _2, \ldots y_t\}$, RNNs aim to model the distribution $P(y_t \mid x_1, x_2, \ldots x_t)$. RNNs model a dynamic system (Figure 1b), where the network state at any t-th time point (*i.e.,* at any t-th position in the SMILES sequence) depends both on the current observation (i.e., the token x_t) and on the previous state of the system at $t-1$. Given an input sequence, generative RNNs are trained to extend this sequence by predicting the next sequence token, defined as $y_t = x_{t+1}$.[30] In this work, we use RNNs with long short-term memory cells (LSTMs),[32] which were proposed to solve the vanishing and exploding gradient problems caused by long sequences and a large network architecture.[33] At any given t-th time step, such a network is described by the following set of equations (Eq. 1):

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
\qquad (1)
$$

where h_t is the hidden state at time t, c_t is the memory cell state at time t, x_t is the input at time t, and h_{t-1} is the hidden state at the previous time step (or the initial hidden state at time $t = 0$); i_t, f_t, o_t are the input, forget, and output gates, respectively. The symbols σ and $\circ$ indicate the sigmoid function and the Hadamard product, respectively; W and b indicate the set of weights and bias values estimated by backpropagation-through-time gradient descent.[34] The memory cell (c_t), encodes the information of the inputs that have been observed until time t. The input gate (i_t) controls the extent to which the new input influences the cell state, the forget gate (f_t) controls what to keep from the previous cell state, and the output gate controls the extent to which the updated cell state is used to compute the new hidden state value (h_t). In this way, important sequence features can be carried along over long time spans, thereby capturing long-distance dependencies.[35] The hidden state value of the LSTM layer (or the last layer of multi-layered LSTM networks, respectively) is the output of the RNN (y_t).

The most common version of RNNs for sequence generation proceeds from left-to-right (forward direction), that is, from $t = 1$ to $t = L$, where L is the length of the SMILES sequence. During training, the first position of the input is filled with a start-of-sentence token, and the last position of the input with an end-of-sequence token. Once the RNN model is trained, the new sequences are generated by (i) inputting the starting token ('G'), and (ii) allowing the model to progressively select the next token, given the respective previous sequence of tokens, until the end token ('E') is generated (Figure 2a). At each step t, the probability for each k-th token to follow the preceding portion of the generated string is computed with a *softmax* function (Eq. 2):

$$P(x_{t+1}=k|x_1,...,x_t)=\frac{\exp(y_t^k)/T}{\sum_{i=1}^{K}\exp(y_t^i)/T} \quad (2)$$

where y_t^k is the model output (logits) for the k-th token at time t (Eq. 1), and i runs over the set K of all tokens. The sampling of tokens is controlled by temperature parameter T. For high temperatures ($T\to\infty$), all tokens have nearly the same probability; the lower the temperature, the more the predicted y_t^k influences the probability of the k-th token. For $T\to 0$, the probability of the token with the highest y_t^k approximates 1. In this present work, 'G' and 'E' were used as start- and end-tokens of SMILES strings, respectively. In what follows, this method will be referred to as "Forward RNN".

Bidirectional methods

Synchronous Forward Backward Recurrent-Neural Network (FB-RNN)

The synchronous Forward Backward RNN (FB-RNN)[29] was developed to predict previous and future words, given an arbitrary starting word within a sentence. We adapted this approach to generate preceding and subsequent portions of SMILES strings, starting from the token at an arbitrary position in the string. In particular, starting from token x_m, the FB-RNN model estimates the two conditional probability distributions in the forward and backward direction as follows (Eq. 3):

$$\begin{aligned}\vec{P}(x_{m+t+1}=k|x_{m-t},...,x_{m+t})&=\frac{\exp(\vec{y}_{m+t}^k)/T}{\sum_{i=1}^{K}\exp(\vec{y}_{m+t}^i)/T}\\ \overleftarrow{P}(x_{m-t-1}=k|x_{m-t},...,x_{m+t})&=\frac{\exp(\overleftarrow{y}_{m-t}^k)/T}{\sum_{i=1}^{K}\exp(\overleftarrow{y}_{m-t}^i)/T}\end{aligned} \quad (3)$$

where $\vec{y}_{m+t}^k$ and $\overleftarrow{y}_{m-t}^k$ are the FB-RNN outputs (Eq. 1) for the k-th token at the t-th time interval in the forward and backward direction, respectively; T is the sampling temperature and i runs over the set K of all tokens. The estimated conditional probabilities are independent. Starting from token x_m, the FB-RNN simultaneously predicts x_{m+t} and x_{m-t} for any t-th time interval by elongating the sequence in the forward and backward direction, respectively (Figure 2c). In our implementation, SMILES string generation starts from the start token ('G') and proceeds in both directions until the end token ('E') is generated on both sides.

Neural Autoregressive Distribution Estimator (NADE)

The Neural Autoregressive Distribution Estimator (NADE)[30] was originally proposed to reconstruct missing values in sequences. Given a sequence of L tokens ($\mathbf{x} = \{x_1, x_2, \ldots x_L\}$) with one missing token in the t-th position (x_t), the NADE model aims to reconstruct the gap by reading the preceding and next parts of the sequence in a forward and backward direction, respectively, and uses this information to replace the missing token. The conditional probability is estimated as follows (Eq. 4):

$$P(x_t=k|\{x_d\}_{\forall d\neq t})=\frac{\exp(\hat{y}_t^k)/T}{\sum_{i=1}^{K}\exp(\hat{y}_t^i)/T} \quad \text{where } \hat{y}_t^i=\overleftarrow{y}_t^i+\vec{y}_t^i \quad (4)$$

where $\overleftarrow{y}_t^i$ is the prediction for the i-th token at the t-th time interval output by the backward RNN (from x_L to x_{t+1}), while $\vec{y}_t^i$ is the prediction of the Forward RNN, which reads the sequence from x_1 to x_{t-1}. The conditional probability is estimated by using both forward and backward information of the sequence. For this present work, NADE was adapted to generating SMILES strings. In particular, a string of tokens representing a missing value ("dummy" token, 'M') was used as the starting sequence, and the model was

used to replace one 'M' token at a time with valid SMILES tokens, either in a predefined or random order until the sequence has no more missing values (Figure 2d).

Bidirectional Molecule Design by Alternate Learning (BIMODAL)

The new Bidirectional Molecule Design by Alternate Learning (BIMODAL) algorithm is inspired by bidirectional RNNs for property prediction,[36] and combines features of both NADE and FB-RNN models. Like NADE, at any *t*-th time step, BIMODAL reads $\mathbf{x} = \{x_m, x_{m+1}, \ldots x_t\}$ along the forward ($x_m \rightarrow x_t$) and backward ($x_t \leftarrow x_m$) direction; like FB-RNN, the SMILES sequence is generated in both directions. However, only one token per step is predicted alternatively on each side by using the left-to-right (forward) and right-to-left (backward) information simultaneously (Figure 2b). Similar to bidirectional RNNs for supervised learning,[36] BIMODAL consists of two RNNs, one for reading the sequence in each direction. The information from the forward and the backward RNN is then combined as given in Eq. 5:

$$y_t = \vec{W}_{hy}\vec{h}_t + \overleftarrow{W}_{hy}\overleftarrow{h}_t + b_{hy} \tag{5}$$

where h_t is the hidden state, W_{hy} is the hidden-to-output weight matrix, and b_{hy} is bias vector, respectively (Eq. 1). In Eq. (5), the arrows indicate the network direction used for the token estimation, *i.e.* forward (→) and backward (←).

In SMILES generation setup, the BIMODAL reads the sequence in both forward and backward directions at each time step *t* (Figure 2b). Then, it generates a new token in either the forward direction ($x_{m+(t-1)/2+1}$, for odd *t* values) or in the backward direction ($x_{m-t/2-1}$ for even values of *t*, Eq. 6):

$$\begin{aligned} P(x_{m+t'+1} = k \,|\, x_{m-t'},\ldots,x_{m+t'}) &= \frac{\exp(y^k_{m+t'})/T}{\sum_{i=1}^{K}\exp(y^i_{m+t'})/T} \quad \text{where } t' = \frac{t-1}{2} \quad (\text{odd } t \text{ values}) \\ P(x_{m-t^*-1} = k \,|\, x_{m-t^*},\ldots,x_{m+t^*}) &= \frac{\exp(y^k_{m-t^*})/T}{\sum_{i=1}^{K}\exp(y^i_{m-t^*})/T} \quad \text{where } t^* = \frac{t}{2} \quad (\text{even } t \text{ values}) \end{aligned} \tag{6}$$

where $y^k_{m+t'}$ and $y^k_{m+t^*}$ are the model output for the *k*-th token at the considered time step (*m-t'* and *m+t**, respectively. String generation starts from the 'G' token and proceeds until the end token ('E') is produced in both directions.

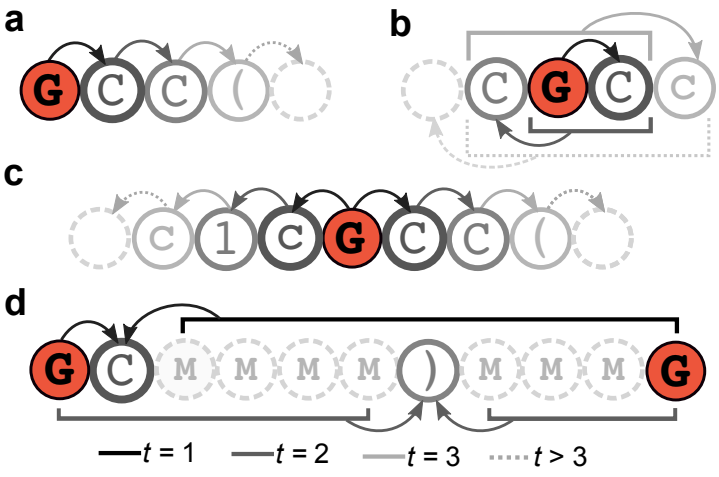

Figure 2. RNN-based methods for SMILES string generation. SMILES generation begins from the start token 'G' and proceeds in a pre-determined direction. (a) Forward RNN: From the start token 'G', new tokens are added from left to right. (b) BIMODAL approach: Tokens are generated in alternating directions at each time step (*t*). The model uses the whole sequence (forward and backward) to generate the next token. (c) Forward-Backward (FB) model: Starting from the 'G' token, two tokens are predicted per time step, one on each side. (d) Neural Autoregressive Distribution Estimator (NADE) approach: Missing "dummy" tokens ('M') are replaced with valid SMILES characters either towards the center of the string or in a random fashion.

Materials and Methods

Data and pre-treatment

All models were trained on canonical SMILES representations of 271,914 bioactive compounds. This training set was compiled from the ChEMBL22[37] database by retaining compounds with annotated $K_{d/I}/IC_{50}/EC_{50}$ <1μM, removing salts and stereochemical information. SMILES strings with less than 34 tokens or more than 74 tokens were omitted. SMILES strings were canonicalized using RDKit[38] (v. 2018.09.2.0) prior to model training.

Model implementation and training

Model architecture

For Forward RNN, NADE and FB-RNN, the network models consisted of five layers (BatchNormalization, LSTM Layer 1, LSTM Layer 2, BatchNormalization, Linear). LSTM layers with either 256 or 512 hidden units (corresponding to a total of 512 or 1024 hidden units, respectively) were implemented and tested. The BIMODAL network was composed of seven layers (BatchNormalization, LSTM Layer 1 - forward, LSTM Layer 1 - backward, LSTM Layer 2 - forward, LSTM Layer 2 – backward, BatchNormalization, Linear). LSTM layers with 128 and 215 hidden units were tested (corresponding to a total of 512 and 1024 hidden units, respectively). A dropout value of 0.3 was used for the output weights in LSTM Layer 1. Details can be found in Supporting Tables S1 and S2.

Model training

Models were trained with the Adam optimization algorithm,[39] using cross-entropy loss (*L*) for performance optimization (Supporting Eq. S1). The cross-entropy loss was computed based on five-fold cross-validation (random partitioning protocol) and fitting (that is, by using all the training data). Models were trained for 10 epochs (one epoch = one pass of all the data points through the network once).

Sequence sampling

SMILES strings were sampled at a sampling temperature of $T = 0.7$, following a procedure for unidirectional RNN sampling from a previous study.[40] For NADE, a sequence of 74 "dummy" missing-value tokens ('M') was generated for further replacement.

Model Evaluation

Evaluation criteria

De novo drug design is a multi-parameter optimization problem.[41] As such, we set out to evaluate several properties and structural characteristics of the molecules produced by each generative method:

1. *Structural uniqueness, validity, and novelty of the generated molecular graphs*. These aspects constitute the primary design goal of any generative model, that is, ensuring that non-redundant, chemically-valid and novel molecules are generated. For the purpose of this study, structural novelty was defined as "not contained in the training set". We evaluated the models based on: (i) uniqueness, calculated as the percentage of unique SMILES strings generated by each model, (ii) validity, calculated as the percentage of chemically-valid and unique (canonical) SMILES strings generated, (ii) novelty, calculated as the percentage of unique and valid molecules that were not included in the training set. The goal was to obtain models that exclusively generate unique, valid, and novel SMILES strings.
2. *Scaffold diversity and novelty*. Exploring novel and diverse scaffolds offers a choice in terms of chemical accessibility and prospects for lead optimization.[42] Thus we evaluated the generative models for their ability to (i) generate a set of molecules with diverse scaffolds, and (ii) produce novel scaffolds (defined

in this study as "not present in the training set"). Bemis-Murcko scaffolds[43] were used for analyzing the scaffold diversity (*ScaffDiv*) and scaffold novelty (*ScaffNov*), defined as:

$$ScaffDiv = n_{scaff} / n \cdot 100, \quad ScaffNov = n^*_{scaff} / n \cdot 100 \tag{9}$$

where n is the number of molecules, n_{scaff} is the number of unique scaffolds, and n^*_{scaff} is the number of unique scaffolds not present in the training set.

3. *Biological and chemical relevance*. Generative models should be able to sample molecules with desired chemical and biological properties, usually resembling the properties of the training set. Here, we compared the property distributions of the designs compared with the training molecules. This was performed by computing the Fréchet ChemNet distance (FCD).[44] FCD values are based on the activation of the penultimate layer of an LSTM model[45] trained to predict bioactivities. The obtained representation thus contains chemical and biological information about the molecular structures, because it is obtained by considering structural (input layer) and biological information (output layer).[44] The lower the FCD between two sets, the closer they are in terms of structural and biological properties.[44]

Each model was trained on a set of 271,914 bioactive molecules from ChEMBL22 for 10 epochs, where the cross-validation loss converged for all cases.

Considered model variants

For bidirectional model training, the starting token can be, in principle, positioned at any position of the SMILES string. We tested two versions of each bidirectional method, based on the position of the starting token ('G') during model training:

1. *Fixed starting position within the molecule*. For BIMODAL and FB-RNN, the starting token was fixed in the center of each SMILES string during training. For NADE, the generation was performed alternatively from the terminal left and terminal right dummy tokens towards the center of the molecule. The fixed-start condition is comparable to the Forward RNN, in which the starting token is always placed in the same position (at the beginning of the sequence).
2. *Random starting position within the molecule*, that is, by randomly placing the starting token within the SMILES string during training. For NADE, the "randomness" was incorporated by performing the replacement of dummy tokens ('M') iteratively in random positions in the sequence.

Additionally, two network sizes (512 and 1024 hidden units in total) were evaluated.

Data augmentation

A feature of bidirectional methods is the possibility to place the starting token in any position of the string during training. We leveraged this feature to perform data augmentation. Data augmentation refers to the artificial incrementation of the training data volume, aimed to increase the model performance.[46] For each training molecule, we generated n repetitions of the canonicalized SMILES strings (n = 5 and n = 10) with a random starting position each. This type of augmentation differs from the approach proposed by Bjerrum,[47] which uses multiple SMILES strings of each molecule.

Software and code

All methods were implemented in Python (v3.6.8), using PyTorch (v1.0.0, https://pytorch.org/). Models were trained on trained on a NVIDIA GeForce GTX 1080 Ti - 256 GB (NVIDIA Corporation, Santa Clara, California, United States). The FCD[44] index was computed using the code made available at URL https://github.com/bioinf-jku/FCD. Statistical tests were performed with MATLAB (vR2018a, The MathWorks, Inc., Natick, Massachusetts, United States). Scaffolds where calculated with RDKit[38] (v2019.03.1).

Results and discussion

Uniqueness, validity and novelty

500 SMILES strings (100 for each cross-validation run) were sampled with each method, and their uniqueness, validity, and novelty were calculated (Table 1). We evaluated the effect of (i) the position of the starting token during the training (fixed *vs.* random), (ii) the network size (512 *vs.* 1024 hidden units), and (iii) data augmentation (5- and 10-fold augmentation).

Table 1. Uniqueness, validity and novelty values (*mean*±*std.dev.*) After 10 training epochs (100 smiles sampled for each five-fold cross-validation run). The highest value for each considered metric is highlighted in boldface.

Model	Starting point	No. Hidden	Data augmentation	Unique (%)	Valid (%)	Novel (%)
Forward	fixed	512	--	100±0	93±2	89±2
BIMODAL	fixed	512	--	99.8±0.5	79±4	79±4
FB-RNN	fixed	512	--	99.4±0.80	51±6	51±6
NADE	fixed	512	--	100±0	19±4	18±4
BIMODAL	random	512	--	100±0	89±3	89±3
FB-RNN	random	512	--	100±0	60±8	60±8
NADE	random	512	--	100±0	7±1	7±1
Forward	fixed	1024	--	100±0	**95±1**	77±4
BIMODAL	fixed	1024	--	99.4±0.8	84±4	81±4
FB-RNN	fixed	1024	--	100±0	53±3	52±3
NADE	fixed	1024	--	100±0	21±2	21±2
BIMODAL	random	1024	--	100±0	89±5	88±4
FB-RNN	random	1024	--	100±0	63±2	62±2
NADE	random	1024	--	100±0	6±3	6±3
BIMODAL	random	512	5x	100±0	91±1	90±1
BIMODAL	random	512	10x	100±0	94±2	**94±1**
FB-RNN	random	512	5x	100±0	63±4	62±4
FB-RNN	random	512	10x	100±0	64±4	64±4
NADE	random	512	5x	100±0	5±2	5±2
NADE	random	512	10x	100±0	6±3	6±3

Effect of starting-token position

With 512 hidden units, all methods generated more than 99% unique molecules, and the Forward RNN reached the highest number of valid and novel molecules (Table 1). In the case of fixed starting point, except for BIMODAL (79% valid and novel molecules), bidirectional generation methods performed only moderately, producing less than 52% valid and novel molecules. Using random starting points boosted the capability of two of the three bidirectional methods to generate valid and novel SMILES, with a 9% to 10% increase for the BIMODAL and FB-RNN (Figure 3). In the case of NADE, decreased validity and novelty were observed, potentially owing to the token replacement at random positions, which led to issues related to SMILES string correctness, *e.g.*, incorrect ring closure. With random starting points, the BIMODAL and Forward RNN methods yielded similar novelty values.

Effect of network size

Most of the bidirectional method did not reach the performance of the Forward RNN with 512 hidden units. Thus, we tested the effect of an increased net size (1024 hidden units) (Table 1). The increased network size had a positive effect on the uniqueness and/or novelty values of all the tested bidirectional methods, with the exception of BIMODAL with random starting points.

Effect of data augmentation

Two levels of data augmentation (five- and ten-fold) were tested. With the exception of NADE, data augmentation led to increased uniqueness and novelty of the generated molecules (Table 1). BIMODAL showed the largest increment of performance, reaching 94% valid and novel molecules with 10-fold augmentation, outperforming the Forward RNN.

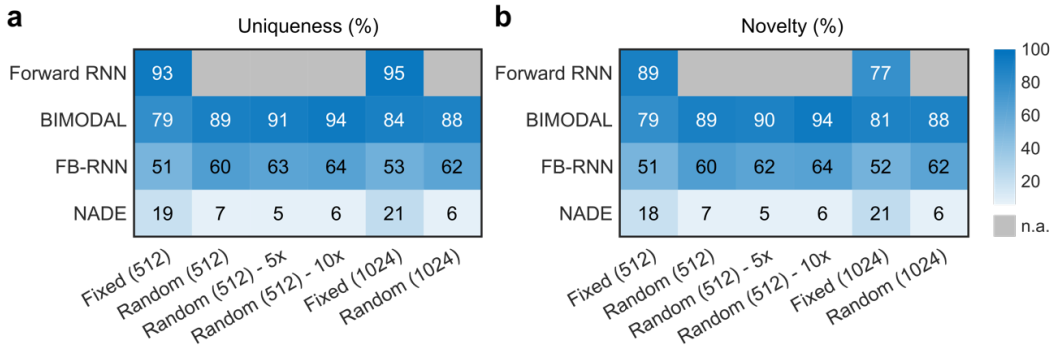

Figure 3. Uniqueness and novelty values (%) obtained by each investigated method (*n.a.* = not available). The generative models were tested with fixed and random starting points, two levels of augmentation (5- and 10-fold), and two network sizes (512 and 1024 hidden units).

Scaffold diversity and novelty

For each model, 30,000 unique and novel SMILES were sampled (5,000 molecules generated in six independent sampling rounds) and analyzed for (i) scaffold diversity, *i.e.,* the percentage of unique scaffolds, and (ii) scaffold novelty, *i.e.*, the percentage of unique scaffolds that are not present in the training molecules (Eq. 9). Five-fold data augmentation was tested, since it resulted the best compromise between the training data volume and compound novelty; NADE with data augmentation was not considered, due to the low validity and novelty of the generated molecules (Figure 3).

Table 2. Scaffold diversity and novelty of the generated molecules (Eq. 9). Different starting points (fixed and random starting point), data augmentation levels, and network sizes were tested (n_h = number of hidden units).

Model	**Starting Point**	**Data augmentation**	**Scaffold diversity** [%]		**Scaffold novelty** [%]	
			n_h = 512	n_h = 1024	n_h = 512	n_h = 1024
Forward	fixed	-	75	72	64	55
BIMODAL	fixed	-	79	81	72	71
BIMODAL	random	-	79	79	72	70
BIMODAL	random	5x	80	79	72	68
FB-RNN	fixed	-	81	84	75	76
FB-RNN	random	-	81	88	75	77
FB-RNN	random	5x	84	89	77	75
NADE	fixed	-	72	73	67	68
NADE	random	-	88	90	85	86

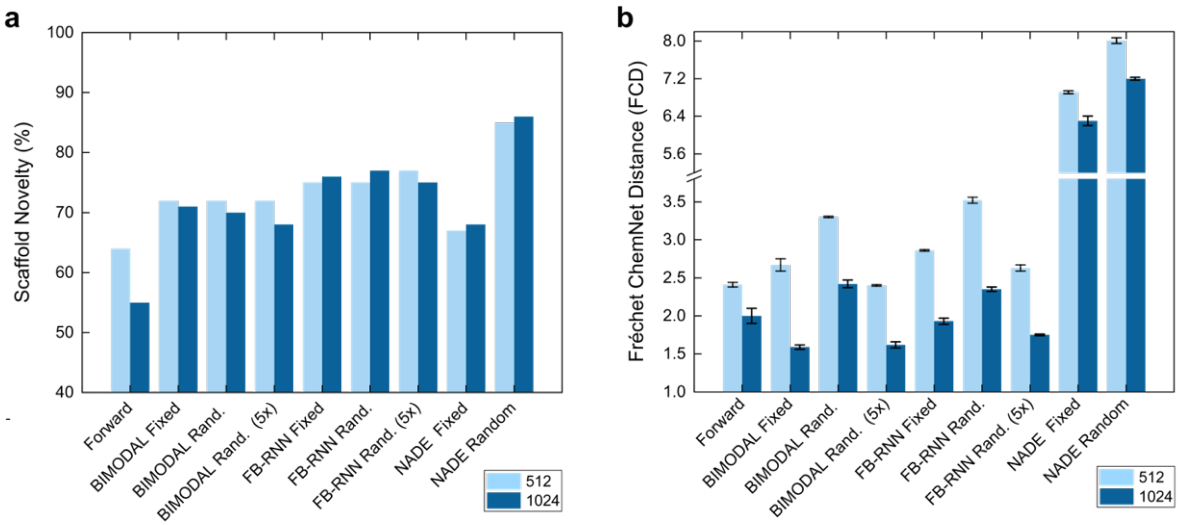

Figure 4. Comparison between generative methods in terms of (a) novelty of the scaffolds (compared to the training set scaffolds – the higher, the better), and (b) Fréchet ChemNet distance (FCD) to the training set molecules (the smaller, the better). Colors represent the net size (512 and 1024 total hidden neurons, corresponding to light gray and dark gray, respectively).

NADE with random replacement yielded the highest values of the investigated metrics. However, many of the scaffolds generated by NADE possess eight-membered rings (Figure 5), which may pose a challenge for synthesis. Of note, the pharmaceutical drugs in DrugBank[48] (v5.0.4) do not contain eight-membered cores. Compared to NADE, the other generative methods produced a more heterogeneous set of scaffolds (Figure 5).

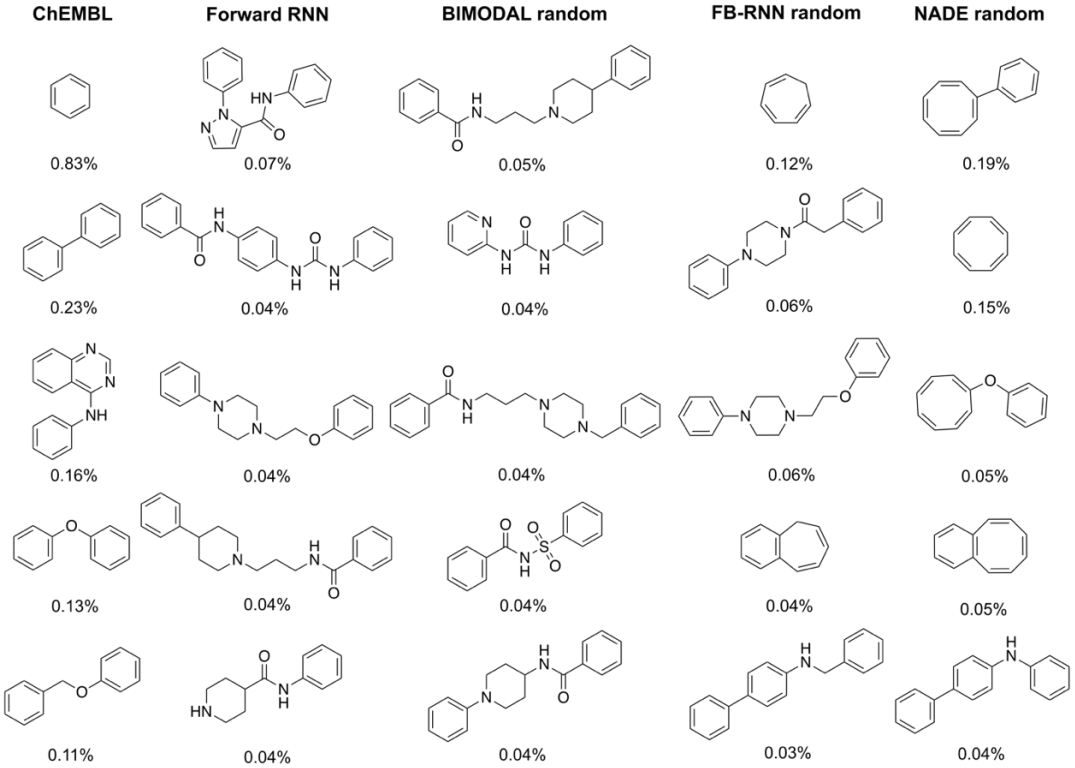

Figure 5. Most frequently generated scaffolds of each investigated model (512 hidden units and no data augmentation), in comparison with ChEMBL22 scaffolds. Only "novel" scaffolds (*i.e.,* scaffolds not present in chembl22) were reported. Numbers indicate the scaffold occurrence in 30,000 de novo designs. The complete list of frequently recurring scaffolds obtained with each method can be found in supporting Table S3.

Biological and chemical relevance

The sampled 30,000 unique and novel SMILES were used to calculate the FCD from the training set molecules. The lower the FCD, the closer the generated molecules are to the training set, in terms of structural and biological properties. Unlike previous studies,[26,49] we only used novel and unique molecules, to not reward models that reproduce the training set.

With 512 hidden units and no data augmentation, the Forward RNN produced the smallest FCD value, followed by BIMODAL, and FB-RNN with fixed starting points (Table 3). NADE showed the highest FCD values – more than twice the values reached by the other methods. The high FCD values indicate that the molecules generated by NADE lie far from the structural and biological domain of the training molecules. This aspect, together with the low validity and novelty of the produced SMILES, and the redundancy of the produced scaffolds, renders NADE sub-optimal for molecular design.

Five-fold data augmentation on BIMODAL and FB-RNN led to FCD values comparable with those of the Forward RNN (no statistically significant difference observed, ANOVA [$\alpha = 0.05$]). With increased network size (1024 hidden units), FCD values decreased for all methods and all the tested cases (Figure 4b). Bidirectional methods experience the largest decrease of FCD values, with BIMODAL reaching significantly smaller FCD values than the Forward RNN ($p<0.001$, ANOVA with Tukey HSD post-hoc analysis). This underscores the potential of BIMODAL to generate molecules closer in the chemical and bioactivity space of training data compared to Forward RNN.

Table 3. FCD values (*mean±std.dev.*) of the generated molecules for each model, with different network size (512 and 1024 hidden units) and different augmentation levels (no augmentation and five-fold augmentation). For each tested number of hidden units, the smallest FCD value is highlighted in boldface.

Model	Starting point	Data augmentation	Number of hidden units	
			512	*1024*
Forward	fixed	-	**2.41±0.03**	2.0±0.1
BIMODAL	fixed	-	2.67±0.08	**1.59±0.03**[a]
FB-RNN	fixed	-	2.86±0.01	1.93±0.04[b]
NADE	fixed	-	6.91±0.03	6.3±0.1
BIMODAL	random	-	3.30±0.01	2.42±0.05
FB-RNN	random	-	3.52±0.04	2.35±0.03
NADE	random	-	8.01±0.06	7.20±0.03
BIMODAL	random	5x	**2.40±0.01**[b]	**1.62±0.04**[a]
FB-RNN	random	5x	**2.63±0.04**[b]	**1.75±0.01**[a]

[a,b]= Results of ANOVA with Tukey HSD post-hoc test to the detect significant differences with the Forward RNN (with the same no. of hidden units, $\alpha = 0.05$). [a] Significantly lower FCD compared to the Forward RNN with the same no. of hidden units ($p<0.001$). [b] No significant difference compared to the Forward RNN model with the same no. of hidden units.

Conclusions and outlook

The results of this study confirm the potential of bidirectional RNNs for *de novo* molecule design. Bidirectional generative methods outperformed the Forward RNN strategy in terms of the applied metrics (*uniqueness, validity, novelty*). The new BIMODAL method seems to be particularly suited for prospective molecule design, judging from the features of the generated chemical entities, *i.e.*, their chemical and biological relevance (as measured by FCD), and their scaffold diversity. FB-RNN yielded moderate results, and NADE proved unsuitable in all of the tested cases. The introduced data augmentation strategy led to more accurate learning of the training data, and increased the novelty of the designs. Taken together, these results advocate the use of bidirectional strategies, in particular BIMODAL, for prospective *de novo* drug design. The current implementation of BIMODAL has long runtimes (Supporting Table S4), owing to the two interacting RNNs. For this reason, alongside the method code, the pre-trained model weights are made available in a GitHub repository (URL: https://github.com/ETHmodlab/BIMODAL), to be used for sampling novel molecules without the need of model re-training.

Acknowledgements

The authors thank Dr. P. Schneider and Dr. J. A. Hiss for technical support. This research was supported by the Swiss National Science Foundation (SNSF, grant no. 205321_182176), the Novartis Forschungsstiftung (FreeNovation: AI in Drug Discovery), and the ETH RETHINK Initiative.

Author contributions

FG and GS conceptualized the study. FG designed the study methodology with contribution from MM and RL. RL and FG implemented the methods and performed the calculations. FG drafted the manuscript. All authors discussed the results and contributed to the manuscript.

References

(1) Dobson, C. M. Chemical Space and Biology. *Nature* **2004**, *432* (7019), 824–828. https://doi.org/10.1038/nature03192.

(2) Schneider, P.; Schneider, G. De Novo Design at the Edge of Chaos. *J. Med. Chem.* **2016**, *59* (9), 4077–4086. https://doi.org/10.1021/acs.jmedchem.5b01849.

(3) Munk, M. E. Computer-Based Structure Determination: Then and Now. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (6), 997–1009. https://doi.org/10.1021/ci980083r.

(4) Fink, T.; Reymond, J.-L. Virtual Exploration of the Chemical Universe up to 11 Atoms of C, N, O, F: Assembly of 26.4 Million Structures (110.9 Million Stereoisomers) and Analysis for New Ring Systems, Stereochemistry, Physicochemical Properties, Compound Classes, and Drug Discovery. *J. Chem. Inf. Model.* **2007**, *47* (2), 342–353. https://doi.org/10.1021/ci600423u.

(5) Wieland, T.; Kerber, A.; Laue, R. Principles of the Generation of Constitutional and Configurational Isomers. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (3), 413–419. https://doi.org/10.1021/ci9502663.

(6) Miyao, T.; Arakawa, M.; Funatsu, K. Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inf.* **2010**, *29* (1–2), 111–125. https://doi.org/10.1002/minf.200900038.

(7) Miyao, T.; Kaneko, H.; Funatsu, K. Inverse QSPR/QSAR Analysis for Chemical Structure Generation (from y to x). *J. Chem. Inf. Model.* **2016**, *56* (2), 286–299. https://doi.org/10.1021/acs.jcim.5b00628.

(8) Miyao, T.; Kaneko, H.; Funatsu, K. Ring-System-Based Exhaustive Structure Generation for Inverse-QSPR/QSAR. *Mol. Inf.* **2014**, *33* (11-12), 764–778. https://doi.org/10.1002/minf.201400072.

(9) Fechner, U.; Schneider, G. Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design. *J. Chem. Inf. Model.* **2007**, *47* (2), 656–667. https://doi.org/10.1021/ci6005307.

(10) Devi, R. V.; Sathya, S. S.; Coumar, M. S. Evolutionary Algorithms for de Novo Drug Design – A Survey. *Appl. Soft Comput.* **2015**, *27*, 543–552. https://doi.org/10.1016/j.asoc.2014.09.042.

(11) Hartenfeller, M.; Zettl, H.; Walter, M.; Rupp, M.; Reisen, F.; Proschak, E.; Weggen, S.; Stark, H.; Schneider, G. DOGS: Reaction-Driven de Novo Design of Bioactive Compounds. *PLOS Comput. Biol.* **2012**, *8* (2), e1002380. https://doi.org/10.1371/journal.pcbi.1002380.

(12) Button, A.; Merk, D.; Hiss, J. A.; Schneider, G. Automated de Novo Molecular Design by Hybrid Machine Intelligence and Rule-Driven Chemical Synthesis. *Nat. Mach. Intell.* **2019**, *1* (7), 307–315. https://doi.org/10.1038/s42256-019-0067-7.

(13) Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. *Learning Internal Representations by Error Propagation*; ICS-8506; California Univ San Diego La Jolla, Inst for Cognitive Science, 1985.

(14) Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci.* **1982**, *79* (8), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554.

(15) Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. *ArXiv Prepr. ArXiv151105644* **2015**.

(16) 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), 268–276. https://doi.org/10.1021/acscentsci.7b00572.

(17) Grisoni, F.; Neuhaus, C. S.; Gabernet, G.; Müller, A. T.; Hiss, J. A.; Schneider, G. Designing Anticancer Peptides by Constructive Machine Learning. *ChemMedChem* **2018**, *13* (13), 1300–1302. https://doi.org/10.1002/cmdc.201800204.

(18) Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *J. Chem. Inf. Model.* **2018**, *58* (6), 1194–1204. https://doi.org/10.1021/acs.jcim.7b00690.

(19) Polykovskiy, D.; Zhebrak, A.; Vetrov, D.; Ivanenkov, Y.; Aladinskiy, V.; Mamoshina, P.; Bozdaganyan, M.; Aliper, A.; Zhavoronkov, A.; Kadurin, A. Entangled Conditional Adversarial Autoencoder for de Novo Drug Discovery. *Mol. Pharm.* **2018**, *15* (10), 4398–4405. https://doi.org/10.1021/acs.molpharmaceut.8b00839.

(20) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models. *ArXiv170510843 Cs Stat* **2017**.

(21) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4* (1), 120–131. https://doi.org/10.1021/acscentsci.7b00512.

(22) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (1), 31–36. https://doi.org/10.1021/ci00057a005.

(23) Ertl, P.; Lewis, R.; Martin, E.; Polyakov, V. In Silico Generation of Novel, Drug-like Chemical Matter Using the LSTM Neural Network. *ArXiv171207449 Cs Q-Bio* **2017**.

(24) Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G. Generative Recurrent Networks for De Novo Drug Design. *Mol. Inf.* **2018**, *37* (1–2), 1700111. https://doi.org/10.1002/minf.201700111.

(25) Bjerrum, E. J.; Threlfall, R. Molecular Generation with Recurrent Neural Networks (RNNs). *ArXiv170504612 Cs Q-Bio* **2017**.

(26) Brown, N.; Fiscato, M.; Segler, M. H. S.; Vaucher, A. C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59* (3), 1096–1108. https://doi.org/10.1021/acs.jcim.8b00839.

(27) Merk, D.; Friedrich, L.; Grisoni, F.; Schneider, G. De Novo Design of Bioactive Small Molecules by Artificial Intelligence. *Mol. Inf.* **2018**, *37* (1–2). https://doi.org/10.1002/minf.201700153.

(28) Merk, D.; Grisoni, F.; Friedrich, L.; Schneider, G. Tuning Artificial Intelligence on the de Novo Design of Natural-Product-Inspired Retinoid X Receptor Modulators. *Commun. Chem.* **2018**, *1* (1), 68. https://doi.org/10.1038/s42004-018-0068-1.

(29) Mou, L.; Yan, R.; Li, G.; Zhang, L.; Jin, Z. Backward and Forward Language Modeling for Constrained Sentence Generation. *ArXiv151206612 Cs* **2015**.

(30) Berglund, M.; Raiko, T.; Honkala, M.; Kärkkäinen, L.; Vetek, A.; Karhunen, J. T. Bidirectional Recurrent Neural Networks as Generative Models. In *Advances in Neural Information Processing Systems 28*; Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc., 2015; pp 856–864.
(31) Jain, L. C.; Medsker, L. R. *Recurrent Neural Networks: Design and Applications*, 1st ed.; CRC Press, Inc.: Boca Raton, FL, USA, 1999.
(32) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9* (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
(33) Hochreiter, S. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *06* (02), 107–116. https://doi.org/10.1142/S0218488598000094.
(34) Werbos, P. J. Generalization of Backpropagation with Application to a Recurrent Gas Market Model. *Neural Netw.* **1988**, *1* (4), 339–356. https://doi.org/10.1016/0893-6080(88)90007-X.
(35) Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *ArXiv14123555 Cs* **2014**.
(36) Schuster, M.; Paliwal, K. K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45* (11), 2673–2681. https://doi.org/10.1109/78.650093.
(37) ChEMBL 22, 2016. 10.6019/CHEMBL.Database.22.
(38) RDKit: Open-source cheminformatics; http://www.rdkit.org. 2019.
(39) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs* **2014**.
(40) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G. Generating Customized Compound Libraries for Drug Discovery with Machine Intelligence (Under Review). **2019**.
(41) Schneider, G. Mind and Machine in Drug Design. *Nat. Mach. Intell.* **2019**, *1* (3), 128–130. https://doi.org/10.1038/s42256-019-0030-7.
(42) Schneider, G.; Schneider, P.; Renner, S. Scaffold-Hopping: How Far Can You Jump? *QSAR Comb. Sci.* **2006**, *25* (12), 1162–1171. https://doi.org/10.1002/qsar.200610091.
(43) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887–2893. https://doi.org/10.1021/jm9602928.
(44) Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. Fréchet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. *J. Chem. Inf. Model.* **2018**, *58* (9), 1736–1741. https://doi.org/10.1021/acs.jcim.8b00234.
(45) Goh, G. B.; Siegel, C.; Vishnu, A.; Hodas, N. Using Rule-Based Labels for Weak Supervised Learning: A ChemNet for Transferable Chemical Property Prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*; KDD '18; ACM: New York, NY, USA, 2018; pp 302–310. https://doi.org/10.1145/3219819.3219838.
(46) Simard, P.; Victorri, B.; LeCun, Y.; Denker, J. Tangent Prop-a Formalism for Specifying Selected Invariances in an Adaptive Network. In *Advances in neural information processing systems*; 1992; pp 895–903.
(47) Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *ArXiv170307076 Cs* **2017**.
(48) Law, V.; Knox, C.; Djoumbou, Y.; Jewison, T.; Guo, A. C.; Liu, Y.; Maciejewski, A.; Arndt, D.; Wilson, M.; Neveu, V.; et al. DrugBank 4.0: Shedding New Light on Drug Metabolism. *Nucleic Acids Res.* **2014**, *42* (D1), D1091–D1097. https://doi.org/10.1093/nar/gkt1068.
(49) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *ArXiv181112823 Cs Stat* **2018**.